Algorithms

An **algorithm** is a set of steps to perform a procedure. For example, the *number guessing game* is a game in which your opponent thinks of a integer n on an interval $[a, b]$, and you try to guess his number with the rule that after every guess he must tell you if your guess was correct, too high, or too low. An algorithm to win this game (in at most $\lfloor \log_2(b-a+1) \rfloor$ guesses, too) is the **binary search algorithm**:

① If $a = b$, then you are done. Otherwise, guess $g = \left\lfloor \frac{a+b}{2} \right\rfloor$.

② If $g = n$, then you are done. If $g > n$, then perform this algorithm on the interval $[a, g-1]$. If $g < n$, then perform this algorithm on the interval $[g+1, b]$.

As we will see, algorithms are a very powerful technique in advanced Olympiad problems, especially given recent IMO trends. In fact, algorithms can be used to prove some of the hardest IMO combinatorics problems ever. In this article, we will discuss greedy algorithms, reduction algorithms, and invariants and monovariants. Some indicators that an algorithm may solve a problem include:

1. A procedure or operation is given. An algorithm would be a helpful organization of a set of allowed steps. Example:

   Example 1.1 (USAMO 2003)**.** At the vertices of a regular hexagon are written six nonnegative integers whose sum is 2003^{2003}. Bert is allowed to make moves of the following form: he may pick a vertex and replace the number written there by the absolute value of the difference between the numbers written at the two neighboring vertices. Prove that Bert can make a sequence of moves, after which the number 0 appears at all six vertices.

2. You seek a construction. An algorithm is a powerful method to construct a configuration satisfying some given conditions. Example:

   Example 1.2. Let Δ be the maximum degree of the vertices of a given graph. Devise a method to color this graph using at most $\Delta + 1$ colors such that no two neighboring vertices are of the same color.

3. A complex object or configuration is given that you want to reduce. Example:

   Example 1.3 (USAMTS 2015)**.** A finite set S of unit squares is chosen out of a large grid of unit squares. The squares of S are tiled with isosceles right triangles of hypotenuse 2 so that the triangles do not overlap each other, do not extend past S, and all of S is fully covered by the triangles. Additionally, the hypotenuse of each triangle lies along a grid line, and the vertices of the triangles lie at the corners of the squares. Show that the number of triangles must be a multiple of 4.

4. You are looking to represent all positive integers n in some given form. Often you are able to represent the items greedily. Example:

   Example 1.4 (Zeckendorf's Theorem)**.** Prove that every positive integer can be written uniquely as the sum of one or more Fibonacci numbers, no two of which are consecutive Fibonacci numbers.

5. You are packing anything. Often you are able to pack the items greedily.

Example 1.5 (Germany 2000)**.** There are stones with a total mass of 9 tons that should be transported by trucks. None of the stones is heavier than 1 ton and each vehicle has a capacity of 3 tons.

Determine the minimum number of necessary trucks such that the stones can be transported at the same time for sure.

6. You are looking for a winning strategy. A winning strategy is, by definition, an *algorithm* to ensure a player of winning a game. Example:

Example 1.6 (Netherlands 2014)**.** Let n be a positive integer. Daniel and Merlijn are playing a game. Daniel has k sheets of paper lying next to each other on a table, where k is a positive integer. On each of the sheets, he writes some of the numbers from 1 up to n (he is allowed to write no number at all, or all numbers). On the back of each of the sheets, he writes down the remaining numbers. Once Daniel is finished, Merlijn can flip some of the sheets of paper (he is allowed to flip no sheet at all, or all sheets). If Merlijn succeeds in making all of the numbers from 1 up to n visible at least once, then he wins. Determine the smallest k for which Merlijn can always win, regardless of Daniel's actions.

We will come back to actually solving these example problems.

GREEDY ALGORITHMS

The **greedy algorithm** is an algorithm that chooses the optimal choice in the short run. A classical example of this is the change-making problem: given a set $\mathcal{S}$ of coin denominations, what is the smallest amount of coins it takes to make a given amount of money M? When $\mathcal{S} = \{1, 5, 10, 25\}$, the United States denominations, we can always apply a greedy algorithm to construct the optimal solution:

① Append the largest element $x \in \mathcal{S}$ such that $x \leq M$ to the solution set.

② Apply the algorithm again on $M - x$.

For example, if we wanted to construct $M = 91$, we would first append 25, then apply the algorithm again on $91 - 25 = 66$. Then we would append 25, apply it again on $66 - 25 = 41$, append 25, apply it again on $41 - 25 = 16$, append 10, apply it again on $16 - 10 = 6$, append 5, apply it again on $6 - 5 = 1$, and append 1 to finish. This gives the set $(25, 25, 25, 10, 5, 1)$. It can be seen that for $\mathcal{S} = \{1, 5, 10, 25\}$, this algorithm produces the optimal solution. However, note that a greedy algorithm does not always give the optimal solution. If $\mathcal{S} = \{1, 3, 4\}$ and $M = 6$, then the greedy algorithm would give $(4, 1, 1)$ when $(3, 3)$ works and is minimal.

Example 2.1 (Binary Number System)**.** Prove that every positive integer can be written uniquely as the sum of one or more distinct powers of 2.

First, we show that each integer *has* a representation by using a greedy algorithm. Choose the largest power of 2, call it 2^k, such that $2^k \leq n$. If $n = 2^k$, then we are already done. Otherwise, we perform the algorithm on $n - 2^k$. Since the powers of 2 have to be distinct, we would have to show that $n - 2^k < 2^k$. However, we already know this to be true, as $n < 2^{k+1}$ from the maximality of k.

Our algorithm devised a method to find the representation of a number. However, when we write our proof, we would use **strong induction**. As a base case, $1 = 2^0$. Assume for all $1 \leq n < 2^k$, for some $k \geq 1$, each n has a representation; we will prove the proposition for all $1 \leq n < 2^{k+1}$. We know that on the

interval $1 \leq n < 2^k$, the representation has largest term at most 2^{k-1}. Adding 2^k to each n on this interval, we find a representation for all $1 + 2^k \leq n < 2^k + 2^k = 2^{k+1}$. Additionally, $n = 2^k$ is its own representation. Therefore, our proposition holds for all $1 \leq n < 2^{k+1}$, so we are done.

To show that this representation is unique, at each step in the induction we have to show that each $2^k \leq n < 2^{k+1}$ *must* have a term of 2^k in its representation. Assuming the contrary, each of these has representation with sum at most $n \leq 2^0 + 2^1 + \cdots + 2^{k-1} = 2^k - 1 < n$, which is a contradiction.

Here is an extension of the techniques in the first problem:

Example 2.2 (Zeckendorf's Theorem)**.** Prove that every positive integer can be written uniquely as the sum of one or more Fibonacci numbers, no two of which are consecutive Fibonacci numbers.

We will use a greedy algorithm to find a representation. Let F_k be the largest Fibonacci number less than n. Then we will reduce it to $n - F_k$, but we have to show that $F_{k-1} > n - F_k$ in order to have the non-consecutive property. However, this is true because $n < F_k + F_{k-1} = F_{k+1}$ from the maximality of k.

Now we use strong induction formalize the proof. As a base case, $1 = F_1$. Assume for all $1 \leq n < F_k$, for some $k \geq 3$, each n has a representation; we will prove the proposition for all $1 \leq n < F_{k+1}$. First, $n = F_k$ is obviously representable. On the interval $1 \leq n < F_{k-1}$, the largest term in the representation must be F_{k-2}, so adding F_k to each term of this interval to find a representation for $1 + F_k < n + F_k < F_{k-1} + F_k = F_{k+1}$. Therefore, we're done.

For the purpose of this article, we need not show that this representation is unique. It is a fun bonus problem, though.

Here is another basic example of a greedy algorithm:

Example 2.3. Let Δ be the maximum degree of the vertices of a given graph. Devise a method to color this graph using at most $\Delta + 1$ colors such that no two neighboring vertices are of the same color.

We shall use the following greedy algorithm: for each vertex, we shall color it with any color that has not been used by any of its neighbors. Since each vertex has at most Δ neighbors, at least one of the $\Delta + 1$ colors has not been used, so such a color will always exist. Thus, we are done.

Now we will look at some harder Olympiad problems involving greedy algorithms:

Example 2.4. Let $A_1, A_2, \ldots, A_n$ be subsets of $\{1, 2, \ldots, n\}$ of size 3. Prove that $\left\lfloor \frac{n}{3} \right\rfloor$ members of $\{1, 2, \ldots, n\}$ can be colored such that each A_i has at least 1 member that is not colored.

There is no condition that each set needs a colored number in it. We may want to **reword the problem** so that we can work backwards: Let $A_1, A_2, \ldots, A_n$ be subsets of $\{1, 2, \ldots, n\}$ of size 3. Prove that $\left\lceil \frac{2n}{3} \right\rceil$ members of $\{1, 2, \ldots, n\}$ can be colored such that each A_i has at least 1 member that is colored.

Why not try a greedy approach? Let A be the set of sets among $A_1, A_2, \ldots, A_n$ without any colored elements at any given moment. At each step, we want to color the number that appears the most in A to reduce $|A|$ by as much as possible.

Suppose that after x applications of this algorithm, all elements of A become disjoint. Then the algorithm would terminate after $x + |A|$ applications in total. However, note that $x \leq \frac{n}{2}$ because each application removes at least 2 sets, and $|A| \leq \frac{n-x}{3}$ because there are $n - x$ numbers remaining to be partitioned into disjoint sets of size 3. Therefore,

$$x + |A| \leq x + \frac{n-x}{3} = \frac{n}{3} + \frac{2x}{3} \leq \frac{n}{3} + \frac{n}{3} = \frac{2n}{3}$$

which implies the algorithm terminates in at most $\left\lceil \frac{2n}{3} \right\rceil$ applications, so we are done.

Example 2.5 (IMO 1983)**.** Is it possible to choose 1983 distinct positive integers, all less than or equal to $100,000$, no three of which are consecutive terms of an arithmetic progression?

Here is some intuition behind the problem: the number $100,000$ seems irrelevant and arbitrarily large, so it would make us think that we could **do it for smaller values and extend it**. We know that we should definitely should start by constructing small sequences using a greedy approach.

We start with the sequence $1, 2$. 3 cannot be added because of $1 \to 2 \to 3$, so we add 4. We add 5. 6 cannot be added because of $2 \to 4 \to 6$, 7 cannot be added because of $3 \to 5 \to 7$, 8 cannot be added because of $2 \to 5 \to 8$, and 9 cannot be added because of $1 \to 5 \to 9$. We add 10. We add 11. We cannot add 12 because of $10 \to 11 \to 12$. We add 13. We add 14. The next number is all the way at 28. Eventually, we get the sequence $\{1, 2, 4, 5, 10, 11, 13, 14, 28, 29\}$. We should analyze when gaps appear, like $2 \to 4$, $5 \to 10$ and $14 \to 28$. One striking observation is that the sequence of numbers after a gap is $4 \to 10 \to 28$. That is just $3^1 + 1 \to 3^2 + 1 \to 3^3 + 1$. Now we know that 3 is important in this problem, specifically powers of 3. This suggests us considering our construction in base 3, but then subtracting 1 from each term. Our construction is thus $1 + \{0, 1, 10, 11, 100, 101, 110, 111, 1000, 1001, 1010, 1011, 1100, 1101, 1110, 1111\}$. With this construction from our original greedy algorithm, we have practically solved the problem.

Consider the set $\mathcal{S}$ of all base 3 numbers less than $100,000$ whose digits are just 0 and 1. We assert that no three distinct numbers $x, y, z \in \mathcal{S}$ satisfy $(1 + y) - (1 + x) = (1 + z) - (1 + y)$, or $x + z = 2y$. Indeed, the digits of $2y$ must be only 0s and 2s. The sum of any two corresponding digits of x and z can either be $0, 1, 2$, which occurs when they are $(0, 0)$, $(0, 1)$, or $(1, 1)$, respectively. Thus, since 1 cannot appear as a digit of $x + z$, we cannot have any two corresponding digits of x and z differ. Thus, $x = z$, which is impossible because x and z distinct. This proves our claim.

Finally, we must show that $|\mathcal{S}| \geq 1983$. Note that $1 + 11111111111_3 < 100,000$, and there are $2^{11} = 2048 > 1983$ elements of $\mathcal{S}$ less than or equal to $1 + 11111111111$. Hence, we are done.

In fact, if we were to replace $100,000$ with N, this greedy approach gives a sequence of size $O\left(N^{\log_3 2}\right)$. This has been improved to $N^{-\sqrt{8 \ln 2 \ln N}(1+o(1))}$ with Behrend's construction.

Example 2.6 (Russia 2005)**.** In a $2 \times n$ array, we have positive reals such that the sum of the numbers in each of the n columns is 1. Show that we can select one number in each column such that the sum of the selected numbers in each row is at most $\frac{n+1}{4}$.

We could start by just selecting just the smallest number in each column. However, what if the smaller numbers always occurred in the same row? Indeed, we find that this greedy algorithm fails if we have

0.4	0.4	0.4	...	0.4
0.6	0.6	0.6	...	0.6

because the sum in the top row would be $0.4n$ which will eventually be greater than $0.25n + 0.25$. But, if we could refine this technique so it would work for both of the rows at the same time, we would be done. We somehow want to select the smallest numbers from one row *and* the smallest numbers from the other row. We can do this by changing the order of the columns such that one row is in **non-decreasing order** while the other row is in non-increasing order: let $a_1, a_2, \ldots, a_n$ be the numbers in the top row such that $a_1 \leq a_2 \leq \cdots \leq a_n$.

a_1	a_2	a_3	...	a_n
$1 - a_1$	$1 - a_2$	$1 - a_3$	...	$1 - a_n$

Choose the largest value of i such that $a_1 + a_2 + \cdots + a_i \leq \frac{n+1}{4}$. We want to show that

$$\frac{n+1}{4} \geq (1 - a_{i+1}) + (1 - a_{i+2}) + \cdots + (1 - a_n) = (n - i) - (a_{i+1} + a_{i+2} + \cdots + a_n)$$

or

$$a_{i+1} + a_{i+2} + \cdots + a_n \geq \frac{3n-1}{4} - i$$

Since $a_1 \leq a_2 \leq \cdots \leq a_{i+1}$, we have

$$\frac{a_1 + a_2 + \cdots + a_{i+1}}{i+1} \leq a_{i+1}$$

and since $a_{i+1} \leq a_{i+2} \leq \cdots \leq a_n$, we have

$$\frac{a_{i+1} + a_{i+2} + \cdots + a_n}{n-i} \geq a_{i+1}$$

Combining the two, we get

$$\frac{a_{i+1} + a_{i+2} + \cdots + a_n}{n-i} \geq \frac{a_1 + a_2 + \cdots + a_{i+1}}{i+1}$$

so

$$a_{i+1} + a_{i+2} + \cdots + a_n \geq (n-i) \cdot \frac{a_1 + a_2 + \cdots + a_{i+1}}{i+1} > (n-i) \cdot \frac{\frac{n+1}{4}}{i+1}$$

from the maximality of i. Finally, we need to show that $\frac{n-i}{i+1}\frac{n+1}{4} \geq \frac{3n-1}{4} - i$. This rearranges to $(n-1)^2 \geq 4i(n-i-1)$ which follows from AM-GM on $2\sqrt{i(n-i-1)} \leq i + (n-i-1) = n-1$, so we are done.

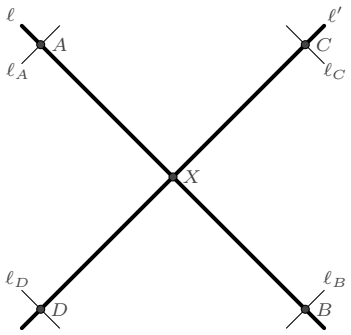
Finally, we will end on a greedy algorithm problem that appeared on the IMO 2014.

Example 2.7 (IMO 2014)**.** A set of lines in the plane is in *general position* if no two are parallel and no three pass through the same point. A set of lines in general position cuts the plane into regions, some of which have finite area; we call these its *finite regions*. Prove that for all sufficiently large n, in any set of n lines in general position it is possible to colour at least $\sqrt{n}$ lines blue in such a way that none of its finite regions has a completely blue boundary.

Surely, we will devise an algorithm to color at least $\sqrt{n}$ lines blue. Whenever you see lines in general position, you should immediately think about the vertices formed by the intersection of two lines. Call two such vertices *adjacent* if they lie on the same line and no other vertex on that line lies between them.

At each step in our algorithm, we are going to color a line blue; the hard part is determining when we can color a line blue. A simple idea to do this is to **mark some *bad* vertices** such that we cannot color a line blue if it passes through a marked vertex. So at each step in our algorithm, we are going to color a line blue and then mark some bad vertices.

Which vertices should we mark after we color a line? Remember, we are marking vertices to ensure that no finite region has an entirely blue boundary. This suggests that we should look at when two blue lines intersect. When we color a line ℓ, for each other blue line ℓ' it intersects, consider the intersection point X. Furthermore, let A and B be the two adjacent vertices to X on ℓ. There is a case in which one of these points does not exist, but in this case we can just pick any arbitrary point on that side. Define C and D analogously on ℓ'. Let ℓ_A, ℓ_B, ℓ_C, and ℓ_D be the other lines passing through A, B, C, and D, respectively. In the diagram below, a thick line represents a blue line:



A greedy approach would suggest marking as few points as possible to ensure that we would have no entirely blue boundaries. Before guessing how to actually do the marking, we can check to see how many vertices we can mark at most to get a tight enough bound. Suppose that our algorithm ends up coloring at most k lines blue. Since we are going to be marking some of A, B, C, or D, each of which already has a blue line passing through it, each marked vertex has exactly one uncolored line passing through it. Thus, the number of marked vertices is at least the number of uncolored lines, $n-k$. If we mark at most m points for each of the k intersections, the number of marked vertices is at most $m\binom{k}{2}$, so

$$m\binom{k}{2} = \frac{mk(k-1)}{2} \geq n-k \implies n \leq \frac{k(m(k-1)+2)}{2}$$

Keeping in mind that we want $n \leq k^2$, we see that at each step we can mark at most $m = 2$ points to keep a tight enough bound. Now we know exactly what to look for! We just need to find an algorithm that marks at most 2 vertices around each blue-blue intersection in a way that none of the finite regions have an entirely blue boundary!

We know that each of AXD, DXB, BXC, and CXA is a part of a region (not necessarily finite). The way to mark the fewest vertices while protecting each of these regions is most likely to mark the opposite vertices like A and B or C and D. If any of the lines ℓ_A, ℓ_B, ℓ_C, or ℓ_D are blue, then we obviously should not mark the vertex at its intersection. This leads us to one possible marking scheme: if ℓ_A and ℓ_B are both uncolored, mark A and B and stop! If ℓ_C and ℓ_D are both uncolored, mark C and D and stop! Otherwise, at least two of ℓ_A, ℓ_B, ℓ_C, and ℓ_D are blue, so just mark any of the other intersections that are not blue-blue, which will be at most 2 points to mark. This algorithm ensures that there will be no finite regions with entirely blue boundaries because there cannot be a step where an entirely blue boundary is created, so we are done.

Although the greedy algorithm is one of the most basic types of algorithms, it can be used to tackle very hard IMO problems. And even when we are not explicitly using a greedy algorithm in a proof, just knowing how to think greedily can often give intuitive and elegant solutions. Despite their ability to frequently give sufficient bounds, greedy algorithms are famous for *not always* working, so be careful when asserting that greedy algorithms give optimal results.

PROBLEMS

Problem 2.1 (Germany 2000)**.** There are stones with a total mass of 9 tons that should be transported by trucks. None of the stones is heavier than 1 ton and each vehicle has a capacity of 3 tons.

Determine the minimum number of necessary trucks such that the stones can be transported at the same time for sure.

Problem 2.2 (Factoradic Number System)**.** Prove that for each positive integer n, there exists a set of integers $a_1, a_2, a_3, \ldots$ such that $a_i \leq i$ for all i and

$$n = a_1 \cdot 1! + a_2 \cdot 2! + a_3 \cdot 3! + \dots$$

Bonus: Prove that this set is unique.

Problem 2.3 (Netherlands 2014)**.** Let n be a positive integer. Daniel and Merlijn are playing a game. Daniel has k sheets of paper lying next to each other on a table, where k is a positive integer. On each of the sheets, he writes some of the numbers from 1 up to n (he is allowed to write no number at all, or all numbers). On the back of each of the sheets, he writes down the remaining numbers. Once Daniel is finished, Merlijn can flip some of the sheets of paper (he is allowed to flip no sheet at all, or all sheets). If Merlijn succeeds in making all of the numbers from 1 up to n visible at least once, then he wins. Determine the smallest k for which Merlijn can always win, regardless of Daniel's actions.

Problem 2.4 (IMOSL 2001, generalised)**.** A set of three nonnegative integers $\{x, y, z\}$ with $x < y < z$ is called *historic* if $\{z - y, y - x\} = \{a, b\}$ for $0 < a < b$. Show that the set of all nonnegative integers can be written as the union of pairwise disjoint historic sets.

Problem 2.5 (Math Prize for Girls 2010)**.** Let S be a set of n points in the coordinate plane. Say that a pair of points is *aligned* if the two points have the same x-coordinate or y-coordinate. Prove that S can be partitioned into disjoint subsets such that (a) each of these subsets is a collinear set of points, and (b) at most $n^{3/2}$ unordered pairs of distinct points in S are aligned but not in the same subset.

Problem 2.6 (IMOSL 2013)**.** Let n be an positive integer. Find the smallest integer k with the following property; Given any real numbers $a_1, \cdots, a_d$ such that $a_1 + a_2 + \cdots + a_d = n$ and $0 \le a_i \le 1$ for $i = 1, 2, \cdots, d$, it is possible to partition these numbers into k groups (some of which may be empty) such that the sum of the numbers in each group is at most 1.

Problem 2.7 (IMO 2014)**.** For each positive integer n, the Bank of Cape Town issues coins of denomination $\frac{1}{n}$. Given a finite collection of such coins (of not necessarily different denominations) with total value at most most $99 + \frac{1}{2}$, prove that it is possible to split this collection into 100 or fewer groups, such that each group has total value at most 1.

Invariants and Monovariants

Example 3.1 (USAMO 1999)**.** Let $a_1, a_2, \dots, a_n$ $(n > 3)$ be real numbers such that

$$a_1 + a_2 + \cdots + a_n \ge n \text{ and } a_1^2 + a_2^2 + \cdots + a_n^2 \ge n^2$$

Prove that $\max\{a_1, a_2, \dots, a_n\} \ge 2$.

Assume for the sake of contradiction that $a_1, a_2, \ldots, a_n < 2$. We will use an algorithm that keeps $s_n = a_1 + a_2 + \cdots + a_n$ invariant and increases $t_n = a_1^2 + a_2^2 + \cdots + a_n^2$. For each $1 \le i \le n-1$, in order, perform the following:

① Let $u = a_i$ and $v = a_{i+1}$.

② Replace a_i with 2 and a_{i+1} with $u + v - 2$.

Clearly, this algorithm preserves s_n. Since a_i increases from $a_i < 2$ to $a_i = 2$, a_{i+1} is decreased. This ensures that at each step, $a_{i+1} < 2$. Furthermore, at each step, we have

$$a_i^2 + a_{i+1}^2 - u^2 - v^2 = 2^2 + (u + v - 2)^2 - u^2 - v^2 = 2(2-u)(2-v) > 0$$

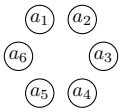
so t_n strictly increases. At the end of the algorithm, $t_n > n^2$ because of the strict increase. Furthermore, $a_n = s_n - 2(n-1) \ge n - 2(n-1) = -n + 2$, so $a_n^2 \le (n-2)^2 = n^2 - 4n + 4$. Finally, we must have

$$n^2 < t_n = 2^2 \cdot (n-1) + n^2 - 4n + 4 = n^2$$

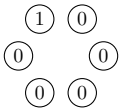
which is a contradiction.

Example 3.2 (USAMO 2003)**.** At the vertices of a regular hexagon are written six nonnegative integers whose sum is 2003^{2003}. Bert is allowed to make moves of the following form: he may pick a vertex and replace the number written there by the absolute value of the difference between the numbers written at the two neighboring vertices. Prove that Bert can make a sequence of moves, after which the number 0 appears at all six vertices.

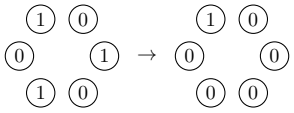
Clearly these numbers are huge, so we need some way of reducing them. Modulo 2 works quite nicely. Let $(a_1, a_2, \ldots, a_6)$ be the numbers on the hexagon:

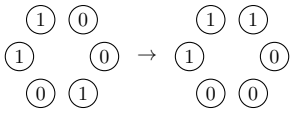


The key observation is that the given operation cannot increase the maximum of a configuration. Our algorithm will work in two steps: first, show that any hexagon with an odd sum can turn into

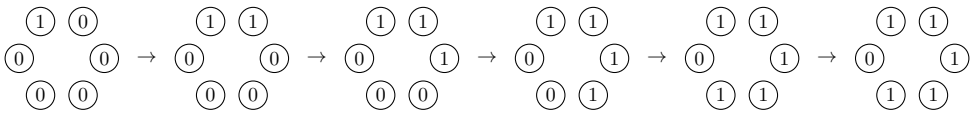


modulo 2. Second, show that such a hexagon can turn into another hexagon with an odd sum and a smaller maximum (or all zeros). For the first part, we have four cases (symmetrically):



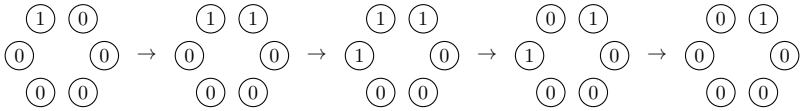


So we can assume a_1 is odd and $a_2, \ldots, a_6$ are even. Now we look at where the maximum m is on the hexagon. When m is even, consider the sequence of moves at $a_2, a_3, \ldots, a_6$ in order. Modulo 2, that is:

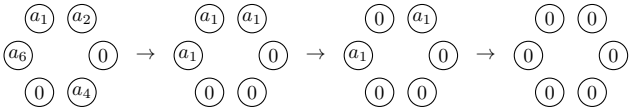


It preserves an odd sum. If $m = a_2, a_3, \ldots, a_5$, then it was replaced by an odd number, so it must have decreased. If $m = a_6$, then note that the difference between its neighbors at the fifth step must be strictly less than m because m is maximal.

Now consider when m is odd, or $m = a_1$. If $a_3 > 0$ then move at a_2, a_6, a_1, and a_6 in order. Modulo 2, that is



Since $a_2' = |m - a_3| < m$, the maximum has reduced. Otherwise, by symmetry, $a_3 = a_5 = 0$, so perform the following:



Example 3.3 (IMOSL 1994)**.** Peter has three accounts in a bank, each with an integral number of dollars. He is only allowed to transfer money from one account to another so that the amount of money in the latter is doubled. Prove that Peter can always transfer all his money into two accounts. Can Peter always transfer all his money into one account?

1. Let the accounts have $0 < A \le B \le C$ dollars each. We want $\min\{A, B, C\} = 0$ at some point, so we should try to create an algorithm to strictly decrease the value of $\min\{A, B, C\}$.

   Consider $(A, B, C) = (7, 39, 52)$. The first thought is to take $(7, 39, 52) \to (14, 25, 52) \to (28, 11, 52)$, but we see that it quickly fails. We want the steps that ensure we can donate the most of B to A. Instead, we could take $(7, 39, 52) \to (14, 32, 52) \to (28, 32, 38) \to (56, 4, 38)$.

   We can try some other random ones, like $(9, 50, 125) \to (18, 41, 125) \to (36, 5, 125)$ or $(15, 29, 61) \to (30, 14, 61)$. One observation we see is that if $B \le 2A$, then the lowest we can go is $(A, B, C) \to (2A, B - A, C)$. Going one level deeper in complexity, on the interval $2A \le B \le 4A$, we have two cases: $2A \le B \le 3A$ gives $(A, B, C) \to (2A, B, C - A) \to (4A, B - 2A, C)$ and $3A \le B \le 4A$ gives $(A, B, C) \to (2A, B - A, C) \to (4A, B - 3A, C)$. One could naturally expect that on $4A \le B \le 8A$, we would need to split this interval into 4 smaller ones. In fact, drawing a binary tree, we can come up with the following table for some small values:

interval	minimum
$[A, 2A]$	$B - A$
$[2A, 3A]$	$B - 2A$
$[3A, 4A]$	$B - 3A$
$[4A, 5A]$	$B - 4A$
$[5A, 6A]$	$B - 5A$
$[6A, 7A]$	$B - 6A$
$[7A, 8A]$	$B - 7A$

The pattern here is clear: we need to prove that B can be reduced to $B \pmod A$. Since we are dealing with powers of 2, we should look at the binary representation of some numbers specifically involving $B \pmod A$. In the case of $(7, 39, 52)$, we have $39 = 7 \cdot 5 + 4 = 111 \cdot 101 + 100$. The corresponds with removing first from B, then from C, then from B, somewhat reminiscent of the number 101.

In fact, letting $B = mA + n$, we assert that the following algorithm will reduce B to $B \pmod A$: consider $m = \overline{m_k m_{k-1} \dots m_1 m_0}$ when written in binary. Then at the ith step, for $0 \le i \le k$, take from B if $m_i = 1$ and take from the C if $m_i = 0$. This is easy to prove because at the beginning of the ith step, there are $2^i A$ dollars in the first account. Thus, after the ith step, we subtract $2^i m_i A$ from B. After the $k+1$ steps, the second account has $B - (2^0 m_0 + 2^1 m_1 + \cdots + 2^k m_k) = B - mA = n$. Since $n = B \pmod A$, we are done.

2. If the accounts have an odd sum, then the final account cannot be doubled, so the answer is no.

PROBLEMS

Problem 3.1 (IMOSL 1989)**.** A natural number is written in each square of an $m \times n$ chess board. The allowed move is to add an integer k to each of two adjacent numbers in such a way that non-negative numbers are obtained. (Two squares are adjacent if they have a common side.) Find a necessary and sufficient condition for it to be possible for all the numbers to be zero after finitely many operations.

Problem 3.2 (China Girls Math Olympiad 2011)**.** There are n boxes $B_1, B_2, \dots, B_n$ from left to right, and there are n balls in these boxes.

1. If there is at least 1 ball in B_1, we can move one to B_2.
2. If there is at least 1 ball in B_n, we can move one to B_{n-1}.
3. If there are at least 2 balls in B_k, $2 \le k \le n-1$ we can move one to B_{k-1}, and one to B_{k+1}.

Prove that, for any arrangement of the n balls, we can achieve that each box has one ball in it.

Problem 3.3 (MEMO 2008)**.** On a blackboard there are $n \ge 2, n \in \mathbb{Z}^+$ numbers. At each step, we select two numbers from the blackboard and replace both of them by their sum. Determine all numbers n for which it is always possible to yield n identical numbers after a finite number of steps.

Problem 3.4 (IMOSL 2013)**.** A crazy physicist discovered a new kind of particle wich he called an imon, after some of them mysteriously appeared in his lab. Some pairs of imons in the lab can be entangled, and each imon can participate in many entanglement relations. The physicist has found a way to perform the following two kinds of operations with these particles, one operation at a time.

1. If some imon is entangled with an odd number of other imons in the lab, then the physicist can destroy it.
2. At any moment, he may double the whole family of imons in the lab by creating a copy I' of each imon I. During this procedure, the two copies I' and J' become entangled if and only if the original imons I and J are entangled, and each copy I' becomes entangled with its original imon I; no other entanglements occur or disappear at this moment.

Prove that the physicist may apply a sequence of much operations resulting in a family of imons, no two of which are entangled.

Reduction Algorithms

Algorithms can be used to reduce a complex configuration to a simple one while preserving its combinatorial function in the problem.

Example 4.1 (Cody Johnson)**.** Consider an ordered set $\mathcal{S}$ of 6 integers. A *move* is defined by the following rule: for each element of $\mathcal{S}$, add either 1 or -1 to it. Show that there exists a finite sequence of moves such that the elements of the resulting set, $\mathcal{S}' = (n_1, n_2, \ldots, n_6)$, satisfy $n_1n_5n_6 = n_2n_4n_6 = n_3n_4n_5$.

The problem is that the elements of $\mathcal{S}$ could be anything, so $n_1n_5n_6 = n_2n_4n_6 = n_3n_4n_5$ could be anything! If there were a way to reduce the possibilities to a small number, it would be helpful...

We see that any set of integers can be reduced to a set with elements only being $0, 1$. We see this is true by the following algorithm: fix one element e and reduce it to zero by applying a sufficient number of moves. Then, to reduce another term, we will apply either an odd or even number of moves. If it is an even number of moves, apply $e + (1 - 1) + (1 - 1) + \cdots = 0$. If it is an odd number of moves, apply $e + 1 + (1 - 1) + (1 - 1) + \cdots = 1$. Thus, we can reduce the other elements of the set while preserving the first element. Therefore, we only need to consider sets with elements in $\{0, 1\}$. Therefore, $n_1n_5n_6$, $n_2n_4n_6$, and $n_3n_4n_5$ are either 0 or 1.

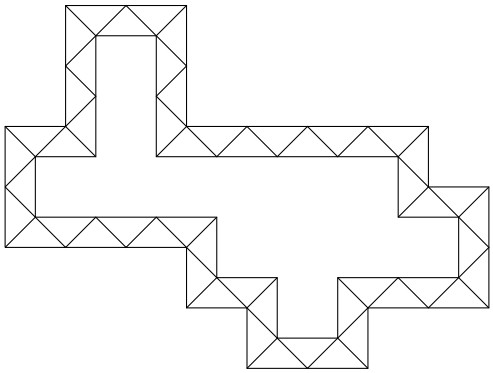
Finally, after using the algorithm to reduce the problem, we can tackle the easy problem. Assume for the sake of contradiction that not all three of the products is equal. This means at least one of $n_1, n_2, \ldots, n_6$ is equal to 0, otherwise each of the products would be 1.

If $n_4 = 0$, then $n_1 = n_5 = n_6 = 1$. Then we could just perform the operation once to get $n_4 = 1$, $n_1 = n_5 = n_6 = 0$ in which case $n_1n_5n_6 = n_2n_4n_6 = n_3n_4n_5 = 0$. The cases of $n_5, n_6 = 0$ can be handled similarly.

If $n_1 = 0$, then we must have either $n_2 = n_4 = n_6 = 1$ or $n_3 = n_4 = n_5 = 1$. In both cases, we have $n_4 = 1$, and either $n_6 = 1$ or $n_5 = 1$. Therefore, perform the operation to get $n_1 = 1$ and $n_4 = n_6 = 0$ or $n_4 = n_5 = 0$, in which case $n_1n_5n_6 = n_2n_4n_6 = n_3n_4n_5 = 0$. The cases of $n_2, n_3 = 0$ can be handled similarly. Therefore, we are done.

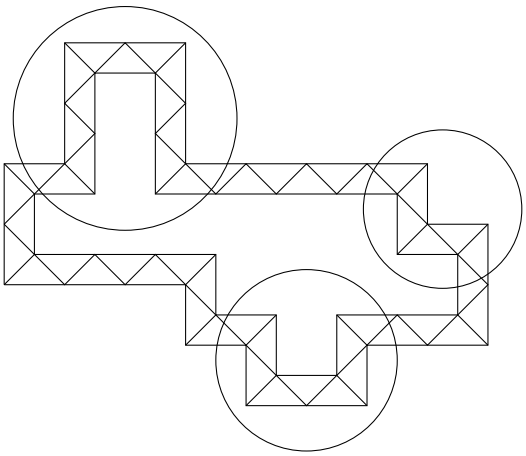
Example 4.2 (USAMTS 2015)**.** A finite set S of unit squares is chosen out of a large grid of unit squares. The squares of S are tiled with isosceles right triangles of hypotenuse 2 so that the triangles do not overlap each other, do not extend past S, and all of S is fully covered by the triangles. Additionally, the hypotenuse of each triangle lies along a grid line, and the vertices of the triangles lie at the corners of the squares. Show that the number of triangles must be a multiple of 4.

First, we can reduce the problem to only connected cycles. This is easy to see because if a connected cycle has 0 triangles modulo 4, then the union of some connected cycles also has 0 triangles modulo 4. Let's draw one example cycle:

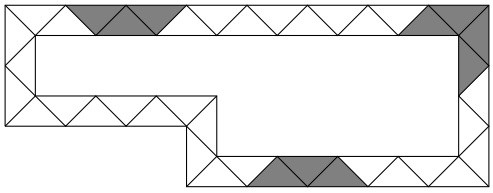


One could try counting the number of times a triangle is pointed in each direction, but that only gives that the total is even. One could try looking at the way each triangle is pointed in the cycle, how each orientation moves is a step up, down, left, or right. These become hard to count and manipulate.

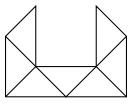
An immediate idea is to use an algorithm to reduce each cycle in total number of triangles while preserving its size modulo 4. From our drawing, some certain parts of the cycle look “easy" to reduce:



Each of the circled regions seems “out of place." We are reassured that this approach seems right because we can reduce them while preserving the size of the cycle modulo 4. Resolving the circled regions, the cycle reduces to:

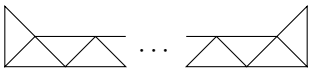


which looks much simpler, yet has the same size modulo 4. Although this worked for this configuration, we need to examine what we actually did in order to formalize our algorithm. In two of the three regions, we took a region in the form of

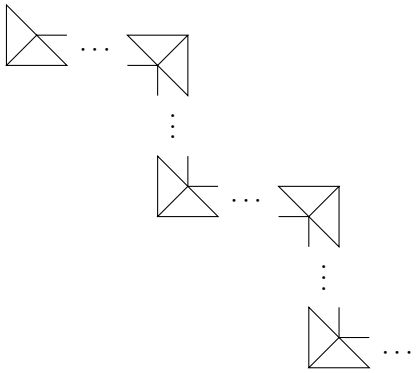


and reduced it to

This seems lucrative, so let us ignore the third case we reduced. We show that every connected cycle of positive size has a region R_n of $n + (n-1) + 2 = 2n + 1$ triangles in the configured as shown in the below diagram:



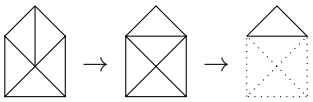
Indeed, if it did not have such a region, it would either be a line or would contain the following region:



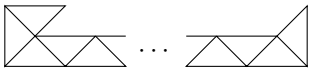
both of which contradict the finiteness of a cycle. Thus, each cycle has such a region.
First, if $n = 1$, then we have the region

which is either the obviously reducible

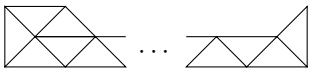
or the less obviously reducible



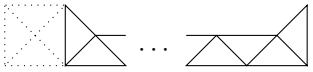
both decreasing the cycle's size by 4. If the region is in the form of



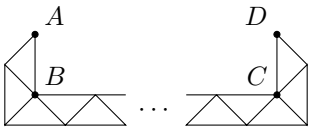
then it must also be in the form of



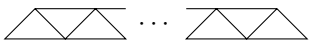
Hence, we can remove the five leftmost triangles and add one to get



reducing the cycle's size by a net 4 triangles. Symmetrically, we only need to consider one more case:



If there is another step of the cycle in rectangle $ABCD$, then the cycle must cross line AD. Therefore, there would be another smaller R_n in $ABCD$. Hence, we can assume without loss of generality that there are no other steps of the cycle in $ABCD$, otherwise we would choose the other smaller region to work with. Finally, we reduce the region by removing the four steps with hypotenuses parallel to AB and CD to get



which also reduces the cycle by 4. We have shown that this algorithm can always be applied if the cycle has a positive size. Noting that cycles of length 1 to 3 do not completely tile a region of unit squares, we are done because the only residue modulo 4 that the size of a cycle can take is 0. This algorithm reduced a large, complex family of cycles to just one case that was easy to solve.

Example 4.3 (China 2010)**.** There are some finite number of cards placed at the points $A_1, A_2, \ldots, A_n$ and O, where $n \geq 3$. We can perform one of the following operations in each step:

1. If there are more than 2 cards at some point A_i, we can remove 3 cards from this point and place one each at A_{i-1}, A_{i+1}, and O. Note that indices are taken modulo n.
2. If there are at least n cards at O, we can remove n cards from O and place one each at $A_1, A_2, \ldots, A_n$.

Show that the if total number of cards is at least $n^2 + 3n + 1$, we can make the number of cards at each vertex at least $n + 1$ after finitely many steps.

The problem with this question is we have no idea how many cards are on each vertices. Fortunately, through the algorithm, we can control the number of cards on each vertex. Indeed, we can always apply the first operation until each vertex has at most 2 cards. The center must have at least $n^2+3n+1-2n = n^2+n+1$ cards in it.

However, once we do this, we are stuck. And when we are stuck, we should try the only thing that works! Applying the second algorithm, each vertex has either 1, 2, or 3 cards. So we will count the number of each occurrence: let there be a vertices with 1 card, b with 2, and c with 3. We have $a + b + c = n$, so the center must have at least

$$n^2 + 3n + 1 - (a + 2b + 3c) = n^2 + 3n + 1 - (2n - a + c) = n^2 + n + 1 + a - c$$

cards. Once we apply the second operation n times, each vertex will have at least $n + 1$ cards and the center will have at least $n^2 + n + 1 + a - c - n^2 = n + 1 + a - c$ cards. Therefore, it suffices to prove that $a \geq c$... or force the inequality to be true! One way to do this is by showing that between any two 3s, there is at least one 1, but there is a problem when the 3s are consecutive.

Fortunately this problem is easily resolved; here is an algorithm to ensure that there are no consecutive 3s: consider a string $x, 3, 3, \ldots, 3, y$ where $x, y \neq 3$. Applying the first operation on the 3s from right to left, we get $x + 1, 1, 2, 2, \ldots, 2, 1, y + 1$. Since we can do this on all chains of consecutive 3s, we can always reach a state with no consecutive 3s.

Finally, we can use another algorithm to eliminate all chains of $x, 3, 2, 2, \ldots, 2, 3, y$. Starting at the 3 and moving to the right, apply the first operation to get $x + 1, 1, 1, \ldots, 1, y + 1$. $x + 1, y + 1 \leq 3$ from the above, so this algorithm ensures that there is a 1 between any two 3s. Hence, $a \geq c$ so we are done.

PROBLEMS

Problem 4.1 (United Kingdom 2011)**.** Initially there are m balls in one bag, and n in the other, where $m, n > 0$. Two different operations are allowed:

1. Remove an equal number of balls from each bag
2. Double the number of balls in one bag.

Is it always possible to empty both bags after a finite sequence of operations? Operation 2 is now replaced with "Triple the number of balls in one bag." Is it now always possible to empty both bags after a finite sequence of operations?

Problem 4.2 (IMO 2010)**.** Each of the six boxes B_1, B_2, B_3, B_4, B_5, B_6 initially contains one coin. The following operations are allowed

1. Choose a non-empty box B_j, $1 \leq j \leq 5$, remove one coin from B_j and add two coins to B_{j+1};
2. Choose a non-empty box B_k, $1 \leq k \leq 4$, remove one coin from B_k and swap the contents (maybe empty) of the boxes B_{k+1} and B_{k+2}.

Determine if there exists a finite sequence of operations of the allowed types, such that the five boxes B_1, B_2, B_3, B_4, B_5 become empty, while box B_6 contains exactly $2010^{2010^{2010}}$ coins.

Problem 4.3 (IMO 2007)**.** In a mathematical competition some competitors are friends. Friendship is always mutual. Call a group of competitors a *clique* if each two of them are friends. (In particular, any group of fewer than two competitiors is a clique.) The number of members of a clique is called its *size*.

Given that, in this competition, the largest size of a clique is even, prove that the competitors can be arranged into two rooms such that the largest size of a clique contained in one room is the same as the largest size of a clique contained in the other room.

SOLUTIONS TO PROBLEMS

Problem 2.1 (Germany 2000)**.** There are stones with a total mass of 9 tons that should be transported by trucks. None of the stones is heavier than 1 ton and each vehicle has a capacity of 3 tons.

Determine the minimum number of necessary trucks such that the stones can be transported at the same time for sure.

Consider 10 stones of mass 0.9 each; clearly these cannot be packed into 3 trucks. Place stones on the truck using a greedy algorithm, noticing that each truck can be packed with greater than 2 tons else we can pack more. Therefore, the mass on the first three trucks is greater 6, meaning the remaining mass of less than 3 tons can be packed on the fourth truck. Thus the answer is 4.

Problem 2.2 (Factoradic Number System)**.** Prove that for each positive integer n, there exists a set of integers $a_1, a_2, a_3, \ldots$ such that $a_i \leq i$ for all i and

$$n = a_1 \cdot 1! + a_2 \cdot 2! + a_3 \cdot 3! + \ldots$$

Bonus: Prove that this set is unique.

We will use a greedy algorithm to find a representation. Let $a_{k-1}(k-1)! < n < k!$, with a_{k-1} maximal. Then we will reduce it to $n - a_{k-1}(k-1)!$, but we have to show that $n - a_{k-1}(k-1)! < (k-1)!$. However, this is true because $n < a_{k-1}(k-1)! + (k-1)! \leq (k-1)(k-1)! + (k-1)! = k!$ from the maximality of k.

Now we use strong induction formalize the proof. As a base case, $1 = 1 \cdot 1!$. Assume for all $1 \leq n < k!$, for some $k \geq 2$, each n has a representation; we will prove the proposition for all $1 \leq n < (k+1)!$. We know that on the interval $1 \leq n < k!$, the representation has largest term at most $(k-1)(k-1)!$. For each $1 \leq a_k \leq k$, adding $a_k \cdot k!$ to each n on this interval, we find a representation for all $1 + a_k \cdot k! \leq n < k! + a_k \cdot k! = (a_k+1) \cdot k!$. Additionally, $n = a_k \cdot k!$ is its own representation. Combining these disjoint intervals, our proposition holds for all $1 \leq n < a_k \cdot k! + k! \leq k \cdot k! + k! = (k+1)!$, so we are done.

To show that this representation is unique, at each step in the induction we have to show that each $k! \leq n < (k+1)!$ *must* have a term of $a_k \cdot k!$ in its representation. Assuming the contrary, each of these has representation with sum at most $n \leq 1 \cdot 1! + 2 \cdot 2! + 3 \cdot 3! + \cdots + (k-1) \cdot (k-1)! = k! - 1 < n$, which is a contradiction.

Problem 2.3 (Netherlands 2014)**.** Let n be a positive integer. Daniel and Merlijn are playing a game. Daniel has k sheets of paper lying next to each other on a table, where k is a positive integer. On each of the sheets, he writes some of the numbers from 1 up to n (he is allowed to write no number at all, or all numbers). On the back of each of the sheets, he writes down the remaining numbers. Once Daniel is finished, Merlijn can flip some of the sheets of paper (he is allowed to flip no sheet at all, or all sheets). If Merlijn succeeds in making all of the numbers from 1 up to n visible at least once, then he wins. Determine the smallest k for which Merlijn can always win, regardless of Daniel's actions.

On the first page, one side must have at least $\lceil \frac{n}{2} \rceil$ numbers on it. Then perform this algorithm with $k-1$ pages and $n - \lceil \frac{n}{2} \rceil = \lfloor \frac{n}{2} \rfloor$ numbers. Considering n in binary, $\lfloor \frac{n}{2} \rfloor$ removes the units digit of n, so the number of applications this algorithm takes to terminate is equal to the number of digits in the binary representation of n, $\lfloor \log_2 n \rfloor + 1$. Note: this algorithm is the binary search algorithm!

To finish, we need to find a construction that requires at least $\lfloor \log_2 n \rfloor + 1$ pages. It suffices to prove that 2^m requires $m+1$ pages. For $1 \leq i \leq m$, on the ith page, write on one side all of the numbers whose ith digit in binary is 1 (where the units digit is the 1st digit). The chosen sides of these m cards must have one number x in common whose digits correspond to which side was chosen on each card, so the number $2^m - 1 - x$ whose digits are opposite to x is not on any of the sides. Note: if $x = 2^m$, then we can take $2^m - 1$. Since one number is not on any of the sides, we require at least $m+1$ pages.

Problem 2.4 (IMOSL 2001, generalised)**.** A set of three nonnegative integers $\{x, y, z\}$ with $x < y < z$ is called *historic* if $\{z - y, y - x\} = \{a, b\}$ for $0 < a < b$. Show that the set of all nonnegative integers can be written as the union of pairwise disjoint historic sets.

For each historic set $\{x, y, z\}$ with $x < y < z$ that we construct, color x red, y green, and z blue. Begin with all nonnegative integers uncolored. We use the following algorithm to construct historic sets, coloring each integer exactly once:

① Consider the smallest uncolored number k.

② If $k + a$ is uncolored, construct the set $\{k, k + a, k + a + b\}$ and color it accordingly. Otherwise, construct the set $\{k, k + b, k + a + b\}$ and color it accordingly.

Now we show that it works. First, $k + a + b$ must be initially uncolored because it is greater than all other colored numbers.

Suppose that $k + b$ is already colored. Then it cannot be red because it is greater than k. It cannot be green because $k + b - a, k + b - b \geq k$. If it were blue, then $k + b - (a + b) = k - a$ must be red. However, since k was uncolored at that moment, our algorithm says that we must color $k - a + a = k$. Therefore, we have reached a contradiction, concluding our proof.

Problem 2.5 (Math Prize for Girls 2010)**.** Let S be a set of n points in the coordinate plane. Say that a pair of points is *aligned* if the two points have the same x-coordinate or y-coordinate. Prove that S can be partitioned into disjoint subsets such that (a) each of these subsets is a collinear set of points, and (b) at most $n^{3/2}$ unordered pairs of distinct points in S are aligned but not in the same subset.

We will use induction on n. As a base case, $n = 0$ is trivial. Now assume that there are at most $k^{3/2}$ pairs of aligned points in S is for all $0 \leq k \leq n - 1$, for some positive integer n.

Consider the largest subset T of S whose points either all have the same x-coordinate or all have the same y-coordinate (in fact, assume without loss of generality it is the same x-coordinate). Let $t = |T|$. For each point $P \in T$, it is part of another set of size at most t with the same y-coordinate, so the number of points in $S \backslash T$ aligned with some point in T is at most $t \cdot t = t^2$. But obviously this can only be at most n pairs because that is the number of points in total. Therefore, it suffices to prove that

$$\min\{t^2, n\} + (n - t)^{3/2} \leq n^{3/2}$$

If $n \leq t^2$, then

$$\min\{t^2, n\} + (n - t)^{3/2} \leq n + (n - \sqrt{n})^{3/2} \leq n^{3/2}$$

for $n \geq 4$, or $t \geq 2$. When $t = 1$, no two points are aligned, the problem statement clearly holds. If $n \geq t^2$, then

$$\min\{t^2, n\} + (n - t)^{3/2} \leq t^2 + (n - t^2)^{3/2} \leq n^{3/2}$$

where the rightmost inequality can be seen from the fact that the inequality holds at $n = t^2$ and the right-hand side always grows faster than the left-hand side (since $n^{3/2}$ is convex and the right-hand side is its translation t^2 units to the right and up).

Problem 2.6 (IMOSL 2013)**.** Let n be an positive integer. Find the smallest integer k with the following property; Given any real numbers $a_1, \cdots, a_d$ such that $a_1 + a_2 + \cdots + a_d = n$ and $0 \leq a_i \leq 1$ for $i = 1, 2, \cdots, d$, it is possible to partition these numbers into k groups (some of which may be empty) such that the sum of the numbers in each group is at most 1.

Consider the set $a_1 = a_2 = \cdots = a_{2n-1} = \frac{n}{2n-1}$. We see that $a_1 + a_2 + \cdots + a_{2n-1} = (2n-1)\frac{n}{2n-1} = n$, and if any two are in the same group then it will have sum at least $\frac{2n}{2n-1} > 1$. Therefore, at least $k = 2n-1$ groups are necessary.

Now suppose that we pack the numbers into as few groups as possible, say $g_1 \leq g_2 \leq \cdots \leq g_m$. Then $g_i + g_j > 1$ for all i, j, else we could combine the groups, reducing m. Furthermore, $g_1 + g_2 + \cdots + g_m = n$. Thus,

$$2n = (g_1 + g_2) + (g_2 + g_3) + \cdots + (g_m + g_1) > 1 + 1 + \cdots + 1 = m$$

so $m \leq 2n - 1$. The answer is $2n - 1$ groups.

Problem 2.7 (IMO 2014)**.** For each positive integer n, the Bank of Cape Town issues coins of denomination $\frac{1}{n}$. Given a finite collection of such coins (of not necessarily different denominations) with total value at most most $99 + \frac{1}{2}$, prove that it is possible to split this collection into 100 or fewer groups, such that each group has total value at most 1.

Replace 100 with m and $99 + \frac{1}{2}$ with $m - \frac{1}{2}$. We are going to greedily put coins into the groups, so we will have to **make some improvements** just to simplify the problem. There is at most 1 coin of denomination $\frac{1}{2k}$ (otherwise combine them into one coin of denomination $\frac{1}{2k} + \frac{1}{2k} = \frac{1}{k}$) and at most $2k$ coins of denomination $\frac{1}{2k+1}$ (otherwise combine them into one coin of denomination $\frac{1}{2k+1} + \frac{1}{2k+1} + \cdots + \frac{1}{2k+1} = 1$). We can also assume that there are no coins of denomination $\frac{1}{1}$ because removing one of them would reduce the problem to $m - 1$ groups of size $(m-1) - \frac{1}{2}$, which is the same problem.

For each k, group all coins of the form $\frac{1}{2k+1}$ and $\frac{1}{2k+2}$. There is at most a total value of $\frac{2k}{2k+1} + \frac{1}{2k+2} < \frac{2k}{2k+1} + \frac{1}{2k+1} = 1$ in each of these groups. For $1 \leq k \leq m-1$, this is $m-1$ groups that take care of the coins of denominations $\frac{1}{3}$ to $\frac{1}{2m}$. We would also have to place $\frac{1}{2}$ into its own group, which is a total of m groups.

Now we will use a greedy algorithm to place the coins of denomination smaller than $\frac{1}{2m}$. Place these coins into whatever groups still have space. We have to prove that this algorithm will always be able to group every coin of denomination smaller than $\frac{1}{2m}$. Assume the contrary, that there is a coin left over. This means all of the other groups are packed with at least $1 - \frac{1}{2m+1}$ coins which is a contradiction because that would mean there is a total value of at least $m\left(1 - \frac{1}{2m+1}\right) > m - \frac{1}{2}$, so we are done.

Problem 3.1 (IMOSL 1989)**.** A natural number is written in each square of an $m \times n$ chess board. The allowed move is to add an integer k to each of two adjacent numbers in such a way that non-negative numbers are obtained. (Two squares are adjacent if they have a common side.) Find a necessary and sufficient condition for it to be possible for all the numbers to be zero after finitely many operations.

Apply the chess board coloring. Let b and w be the sums in the black and white squares. We see that $b - w$ is invariant, and at the end $b - w = 0$, so $b = w$ is necessary.

Now we prove it is sufficient. Consider the numbers in squares (x, y, z) adjacent in that order, and we want to find an algorithm to make $x = 0$. If $x \leq y$, apply $(x, y, z) \to (0, y - x, z)$. If $x \geq y$, apply $(x, y, z) \to (x, x, z + x - y) \to (0, 0, z + x - y)$. We can apply this algorithm first to all the rows from left to right, then to all the columns from top to bottom to reduce the problem to a 2×2 square.

In the 2×2 square, we can make at least one number in each row 0. If they are in the same column, then $b = w$ implies that the remaining numbers are equal, so eliminate them. If they are in opposite columns, then one of b, w is 0, so all entries must be 0.

Problem 3.2 (China Girls Math Olympiad 2011)**.** There are n boxes $B_1, B_2, \ldots, B_n$ from left to right, and there are n balls in these boxes.

1. If there is at least 1 ball in B_1, we can move one to B_2.
2. If there is at least 1 ball in B_n, we can move one to B_{n-1}.
3. If there are at least 2 balls in B_k, $2 \le k \le n-1$ we can move one to B_{k-1}, and one to B_{k+1}.

Prove that, for any arrangement of the n balls, we can achieve that each box has one ball in it.

We use the following algorithm:

① Perform the second and third operation until they can no longer be performed. This leaves boxes $B_2, B_3, \ldots, B_{n-1}$ with at most 1 ball and B_n with at most 0 balls.

② Consider the smallest k such that B_k has 0 balls. That is, $(B_1, B_2, B_3, \ldots, B_{k-1}, B_k) = (x, 1, 1, \ldots, 1, 0)$.

③ Apply the first operation to get $(x-1, 2, 1, \ldots, 1, 0)$.

④ Apply the third operation on $B_2, B_3, \ldots, B_{k-1}$ to end up with $(x, 1, 1, \ldots, 1, 1, 0, 1)$.

⑤ Repeat ② and ③ enough to get $(x, 1, 1, \ldots, 1, 1, 0) \to (x, 1, 1, \ldots, 1, 0, 1) \to (x, 1, 1, \ldots, 0, 1, 1) \to \ldots (x, 0, 1, 1, \ldots, 1, 1)$.

⑥ Apply the first step to get $(x-1, 1, 1, 1, \ldots, 1, 1)$.

⑦ Apply ② through ⑥ to get the desired result.

The only issue is proving that ① always terminates. To do this, weight the boxes to create a monovariant, then show that the weighted sum is bounded above. In fact, any application of the second or third operation strictly increases the sum

$$B_0 2^n + B_1 2^{n-1} + B_2 2^{n-2} + \cdots + B_{n-1} 2^1 + B_n 2^0$$

because $2(B_{n-1}+1) + (B_n - 1) = 2B_{n-1} + B_n + 1 > 2B_{n-1} + B_n$ and $2^{n-i+1}(B_{i-1}+1) + 2^{n-i}(B_i - 2) + 2^{n-i-1}(B_{i+1}+1) = B_{i-1}2^{n-i+1} + B_i 2^{n-i} + B_{i+1}2^{n-i-1} + 2^{n-i+1} - 2 \cdot 2^{n-i} + 2^{n-i-1} > B_{i-1}2^{n-i+1} + B_i 2^{n-i} + B_{i+1}2^{n-i-1}$ from $2^{n-i+1} - 2 \cdot 2^{n-i} + 2^{n-i-1} = 2^{n-i-1} > 0$.

Finally, since

$$B_0 2^n + B_1 2^{n-1} + B_2 2^{n-2} + \cdots + B_{n-1} 2^1 + B_n 2^0 \le n \cdot 2^n$$

this algorithm must terminate, so we are done.

Problem 3.3 (MEMO 2008)**.** On a blackboard there are $n \ge 2, n \in \mathbb{Z}^+$ numbers. At each step, we select two numbers from the blackboard and replace both of them by their sum. Determine all numbers n for which it is always possible to yield n identical numbers after a finite number of steps.

For odd n, we quickly notice the counterexample $(2, 2, \ldots, 2, 1)$. Let m be the maximum among the numbers. The invariant here is the parity of the number of terms that are equal to m, which must end up even.

We assert that all even numbers work. Let $a_1, a_2, \ldots, a_{2n}$ be the numbers on the board. Since n is even, we can group the terms into pairs: $(a_1, a_2), (a_3, a_4), \ldots, (a_{2n-1}, a_{2n})$. Furthermore, let $b_1, b_2, \ldots, b_{2n}$ be $a_1, a_2, \ldots, a_{2n}$ divided by the largest power of 2 that divides all terms. Then, perform the following algorithm:

① For each $1 \le i \le n$, replace (a_{2i-1}, a_{2i}) with $(a_{2i-1} + a_{2i}, a_{2i-1} + a_{2i})$. Now $a_{2i-1} = a_{2i}$ for $1 \le i \le n$.

② For each $1 \le i \le n$, if $a_{2i-1} = a_{2i}$ is odd, replace (a_{2i}, a_{2i}) with $(2a_{2i}, 2a_{2i})$. Repeat this step until each of $a_1, a_2, \ldots, a_{2n}$ has the same common power of 2, i.e., $b_1, b_2, \ldots, b_n$ are all odd. This step can be referred to as "dividing" a pair (b_{2i-1}, b_{2i}) by 2.

③ Add the largest pair with the smallest pair.

④ Return to step 2.

This algorithm may look somewhat complex, but it is actually fairly easy:

a_1	a_2	a_3	a_4	a_5	a_6	b_1	b_2	b_3	b_4	b_5	b_6	
3	4	4	2	3	3	3	4	4	2	3	3	initial
7	7	6	6	6	6	7	7	6	6	6	6	①
14	14	6	6	6	6	7	7	3	3	3	3	②
20	20	20	20	6	6	10	10	10	10	3	3	③
20	20	20	20	12	12	5	5	5	5	3	3	②
32	32	20	20	32	32	8	8	5	5	8	8	③
32	32	160	160	32	32	1	1	5	5	1	1	②
192	192	192	192	32	32	6	6	6	6	1	1	③
192	192	192	192	256	256	3	3	3	3	1	1	②
256	256	192	192	256	256	4	4	3	3	4	4	③
256	256	768	768	256	256	1	1	3	3	1	1	②
1024	1024	1024	1024	256	256	4	4	4	4	1	1	③
1024	1024	1024	1024	1024	1024	1	1	1	1	1	1	②

Now we have to prove that this algorithm terminates. Consider a set of $(b_1, b_2, \ldots, b_n)$ with all odd elements and $b_1 + b_2 + \cdots + b_n > 6$ (else we are already done). It's clear that we can such a state from any given position. After applying ③ and ②, we assert that $b_1 + b_2 + \cdots + b_n$ has decreased. Without loss of generality, let (b_1, b_2) be the maximum pair and (b_3, b_4) the minimum pair. After the operation, we have $b_1', b_2', \ldots, b_n'$. We know that for some k, since $b_1 + b_3$ is even,

$$b_1' + b_3' = b_1' + b_3' = \frac{b_1 + b_3}{2^k} \leq b_1 + b_3$$

and similarly with $b_2 + b_4$. Therefore, the sum always decreases, so it will always decrease to 6.

Problem 3.4 (IMOSL 2013)**.** A crazy physicist discovered a new kind of particle wich he called an imon, after some of them mysteriously appeared in his lab. Some pairs of imons in the lab can be entangled, and each imon can participate in many entanglement relations. The physicist has found a way to perform the following two kinds of operations with these particles, one operation at a time.

1. If some imon is entangled with an odd number of other imons in the lab, then the physicist can destroy it.
2. At any moment, he may double the whole family of imons in the lab by creating a copy I' of each imon I. During this procedure, the two copies I' and J' become entangled if and only if the original imons I and J are entangled, and each copy I' becomes entangled with its original imon I; no other entanglements occur or disappear at this moment.

Prove that the physicist may apply a sequence of much operations resulting in a family of imons, no two of which are entangled.

Let the imons be the vertices of a graph, and connect two vertices of the graph G with an edge if the two are entagled. We assert that we can always decrease the chromatic number $\chi(G)$ (the minimum number of colors to color the vertices of a graph such that no two adjacent vertices are the same color) with an algorithm:

① Apply the first operation until all vertices have an even degree to get G'. Clearly $\chi(G') \leq \chi(G)$.

② Apply the second operation to get G''. We can still color this graph with $\chi(G')$ colors: each instance a vertex on G' was colored with c_i (out of the $\chi(G')$ colors $c_1, c_2, \ldots, c_{\chi(G')}$), we can color the copied vertex with c_{i+1}, for example. The degree of each vertex of G'' is thus odd.

③ Delete the vertices of some color c one by one. Since no two vertices of c are connected, the degrees don't change as we perform this. The resulting graph has $\chi(G'') = \chi(G') - 1 \leq \chi(G) - 1$.

Problem 4.1 (United Kingdom 2011)**.** Initially there are m balls in one bag, and n in the other, where $m, n > 0$. Two different operations are allowed:

1. Remove an equal number of balls from each bag
2. Double the number of balls in one bag.

Is it always possible to empty both bags after a finite sequence of operations? Operation 2 is now replaced with "Triple the number of balls in one bag." Is it now always possible to empty both bags after a finite sequence of operations?

1. Perform the following algorithm:

   ① If $m = n$ remove all the balls from each bag.
   ② Assume without loss of generality $m > n$. Remove $n - 1$ balls from each bag.
   ③ Double the bag with 1 ball in it to get 2 balls.
   ④ Remove 1 from each bag.
   ⑤ Repeat ③ and ④ until both bags have 1 ball.
   ⑥ Remove 1 ball from both bags.

   It is clear that this algorithm terminates.

2. Let $m + n \equiv 1 \pmod 2$. Then $(m - k) + (n - k) \equiv m + n \equiv 1 \pmod 2$ and $3m + n \equiv m + n \equiv 1 \pmod 2$. Thus, $m + n \neq 0$, so it is actually *im*possible.

Problem 4.2 (IMO 2010)**.** Each of the six boxes B_1, B_2, B_3, B_4, B_5, B_6 initially contains one coin. The following operations are allowed

1. Choose a non-empty box B_j, $1 \leq j \leq 5$, remove one coin from B_j and add two coins to B_{j+1};
2. Choose a non-empty box B_k, $1 \leq k \leq 4$, remove one coin from B_k and swap the contents (maybe empty) of the boxes B_{k+1} and B_{k+2}.

Determine if there exists a finite sequence of operations of the allowed types, such that the five boxes B_1, B_2, B_3, B_4, B_5 become empty, while box B_6 contains exactly $2010^{2010^{2010}}$ coins.

Let $N = 2010^{2010^{2010}}$. Let the boxes have $(a_1, a_2, \ldots, a_6)$ coins at a given moment. We assert that $(1, 1, 1, 1, 1, 1)$ can be turned into $(0, 0, 0, 0, 0, N)$. To do this, we shall prove two statements first.

We assert that $(x, 0, 0)$ can go to $(x - y, 2^y, 0)$. Indeed, $(x, 0, 0) \to (x - 1, 2^1, 0)$ and

$$(x - y, 2^y, 0) \to (x - y, 2^y - 1, 2) \to (x - y, 2^y - 1, 4) \ldots (x - y, 0, 2^{y+1}) \to (x - y - 1, 2^{y+1}, 0)$$

In particular, $(x, 0, 0)$ can go to $(0, 2^x, 0)$.

Now let $T_n = \underbrace{2^{2^{\cdots}}}_{n}$. We assert that $(x-y,0,0,0)$ can go to $(0,T_y,0,0)$. Indeed, $(x,0,0,0) \to (x-1,2,0,0)$ and

$$(x-y,T_y,0,0) \to (x-y,0,2^{T_y} = T_{y+1},0) \to (x-y-1,T_{y+1},0,0)$$

In particular, $(x,0,0,0)$ can go to $(0,T_x,0,0)$.

To solve the original problem, consider

$$(1,1,1,1,1,1) \to (1,1,1,1,0,3) \to (1,1,1,0,3,0) \to (1,1,0,3,0,0) \to (1,0,3,0,0,0)$$
$$\to (0,3,0,0,0,0) \to (0,0,16,0,0,0) \to (0,0,0,T_{16},0,0)$$

Notice that

$$2010^{2010^{2010}} < (2^{11})^{2010^{2010}} = 2^{11\cdot 2010^{2010}} < 2^{2010^{2011}} < 2^{(2^{11})^{2011}} = 2^{2^{11\cdot 2011}} < 2^{2^{2^{15}}} < T_{16}$$

so apply the second operation on the last 3 boxes to reach $(0,0,0,N/4,0,0)$. Finally, apply the first operation sufficiently to reach

$$(0,0,0,N/4,0,0) \to (0,0,0,0,N/2,0) \to (0,0,0,0,0,N)$$

Problem 4.3 (IMO 2007)**.** In a mathematical competition some competitors are friends. Friendship is always mutual. Call a group of competitors a *clique* if each two of them are friends. (In particular, any group of fewer than two competitiors is a clique.) The number of members of a clique is called its *size*.

Given that, in this competition, the largest size of a clique is even, prove that the competitors can be arranged into two rooms such that the largest size of a clique contained in one room is the same as the largest size of a clique contained in the other room.

Let the two rooms be A and B; let $c(X)$ be the maximum size of a clique in room X. We will use an algorithm to create such a configuration:

① Let C be one of the the largest cliques with $|C| = 2n$. Send the members of C to A and the others to B. We have $c(A) \geq c(B)$.

② As long as $c(A) > c(B)$, send one person from A to B. $c(A)$ has decreased by exactly 1 and $c(B)$ has increased by at most 1. Thus, this step leaves $c(A) \leq c(B) \leq c(A)+1$. Also, $c(A) = |A| \geq n$ else there will be $n+1$ members of C in B and $n-1$ members of C in A, implying $c(B) - c(A) = (n+1)-(n-1) = 2 > 1$, contradiction.

If $c(A) = c(B)$, then we are done, so $c(B) = c(A)+1$. If there is a member of C in B not in a largest clique of B, then move him back to A to finish. Otherwise, for each largest clique in B move 1 competitor *who is not a member of* C back to A. This decreases $c(B)$ by 1, so to finish, we must show that $c(A)$ did not change after this operation.

Clearly $c(A)$ did not decrease. Let X and Y be the sets members of C in A and B, respectively. Let Z be the largest clique in A.

Note that $c(B)+1 > n$, and n is greater than the number of members of C in B. Therefore, every clique of largest size in B must contain Y.

Finally, we assert that $Z \cup Y$ is a clique. Each competitor in Z is either a member of C, in which case it knows all members of Y, or it was a member of a clique containing Y but then moved into A, in which case it knows all members of Y. Therefore, $Z \cup Y$ is a clique, so

$$|C| \geq |Z \cup Y| = |Z| + |Y| = |Z| + (|C| - |X|)$$

so $|X| \geq |Z|$. This proves that $c(A)$ could not increase in the operation of sending some members back, so the largest cliques in each of the two rooms have the same size.