

A. Sum Game

Author: Adrian Miclaus
Solved by: 78/98
First to solve: *pretests passed (4)*

For any subset $S \subseteq \{1, \dots, N\}$ define its (unsigned) subset sum modulo M : $f(S) = \sum_{i \in S} a_i \pmod{M}$. If there exist two distinct subsets $S \neq T$ with $f(S) = f(T)$, then $\sum_{i \in S \setminus T} a_i - \sum_{i \in T \setminus S} a_i \equiv f(S) - f(T) \equiv 0 \pmod{M}$. Ana chooses “+” on $S \setminus T$, “−” on $T \setminus S$, and erases the rest. Since $S \neq T$, the remaining set is non-empty. Thus, Ana can win iff there are two distinct subsets with equal subset-sum modulo M . There are 2^N subsets, but only M possible residues modulo M . If $2^N > M$, then by the pigeonhole principle two distinct subsets collide, so Ana can always form a signed sum divisible by M . If $2^N \leq M$, Bob chooses $a_i = 2^{i-1}$, $1 \leq i \leq N$. Then every subset sum as an integer lies in $[0, 2^N - 1]$ and is unique, hence also unique modulo M because $2^N - 1 < M$. So, no two distinct subsets have the same residue, and in particular no nonempty subset sums to 0 modulo M . In this case, Ana cannot win.

B. AND Reconstruction

Author: Pavle Martinović
Solved by: 45/69
First to solve: *Sigma--1*

First, it's easy to notice that if k is a valid solution, then so is $k - 1$ because the AND of an interval of range k can be obtained as an AND of two ANDs of intervals of length $k - 1$ contained inside it. Also $k = 1$ is always a solution since that's just the array. Because of this, and because all bits are independent, we can solve all bits independently and taking the minimum to obtain the full solution. So we can solve the problem for just 1 bit and every element is either 0 or 1.

The first insight is that if there are three consecutive zeros, then the answer is always 1. The reason for this is because the middle zero can be replaced with a one and not change any AND of k consecutive elements for $k > 2$, which obviously prevents reconstruction.

Now if there are no three consecutive zeros, the blocks of zeros are either of length 1 or 2, while between them, there are some blocks of ones. Let a be the shortest block of ones. We claim this is the solution. If k is larger than a , then the shortest block of ones can be replaced with all zeros and will not change any AND of a range of length k , so construction is impossible. On the other hand, if k is equal to a , then any block of ones will be "identified" by a block of size k , and moving one-by-one we can find the first zero to the left and to the right of each block. However, since there are at most 2 zeros in a block of zeros, this allows us to identify all zeros! This means that this uniquely determines the array.

C. XOR-Excluding Sets

Author: Pavle Martinović
Solved by: 10/15
First to solve: *Sigma=-1*

The first insight is that we will actually want to calculate the number of sets that do include their own XOR and subtract that number from the total number of sets. A set includes its own XOR if and only if it's gotten by adding exactly one element to a set with XOR equal to 0. Now we are interested in the sum "how many times is an element outside a set of XOR 0".

Firstly, we are interested in the number of sets with XOR 0. This is well known if we have a vector basis: $2^{N-\text{basis_size}}$.

Lemma: Let Φ be the family of all subsets of X of A with XOR 0. Every element $x \in A$ is in either 0 or $\frac{|\Phi|}{2}$ elements of Φ .

Proof: Suppose that an element x is in a subset B of A with XOR 0. Then we can construct a bijection between the XOR 0 subsets that include x and those that don't - just take the set XOR of that set and B ! Indeed, this will keep the XOR 0, remove x if it is in B , and add it in if it is not. This means that every element is in either 0 XOR 0 subsets (if B does not exist) or precisely half.

Now, to calculate everything, we just need the number of elements that are precisely in 0 XOR 0 sets (with a basis everything can be calculated with this info). The key insight here is that those elements have to be in the basis of the set, because all others can be represented using the elements that are in the basis so taking that representation plus that element generates at least one such set. Furthermore, the only way to have an element with no XOR 0 subset containing it is for it to never be used in the representation of any element of the set outside the basis, because if such an element exists, taking the set containing it and its representation would produce one such set.

So to fully solve this, we just add elements one by one and maintain the basis. When we represent the elements outside the basis, we also memorize the number of elements from the basis that are never used.

D. Two Options

Author: Roman Bilyi
Prepared by: Yarema Stiahar
Solved by: 22/31
First to solve: *2Popici1Arici*

Let's assume some value v there are at least 2 triplets with that value (a_1, b_1, v) and (a_2, b_2, v) . If all values a_1, b_1, a_2, b_2 are distinct, the answer is 0. WLOG, $a_1 = b_2 = x$, then should hold $p_x = v$. So we can find positions of all values that appear more than once (or the answer is 0). After that, while we have triplets (i, j, v) such that either position i or j is used, we can set value v to another one. After that we have left only triplets with unique values and no positions solved.

For each triplet (a, b, v) let's create undirected edge between nodes a and b . We must assign each value to one of its possible nodes such that each node receives at most one value (some nodes may receive values that weren't present in any triplet). Let's say that we must direct edges such that for each node at most one edge points towards it. Note that actual values v don't matter.

We decompose the graph into connected components. Let a component have V_c vertices and E_c edges. Since each vertex can hold only one value, we must have $E_c \leq V_c$. If $E_c > V_c$ for any component, the answer is 0.

We have the following cases:

- Tree ($E_c = V_c - 1$)

We have $V_c - 1$ values to place in V_c spots. Exactly one spot will remain empty. Any of the V_c vertices can be the empty one. Once the empty vertex is chosen, the orientation of all edges is forced.

Number of ways: V_c .

- Unicyclic ($E_c = V_c$)

We must fill all V_c spots with V_c edges. This is only possible by orienting the unique cycle in one of 2 directions (clockwise or counter-clockwise). The edges of trees attached to the cycle are forced to point away from the cycle.

Number of ways: 2.

Let K be the number of values that were completely unconstrained (not presented in any triplet). These values will fill the K positions left empty by the tree components. The total number of good permutations is:

$$\text{Ans} = \left(\prod_{\text{components } c} \text{Ways}(c) \right) \times K! \pmod{10^9 + 7}$$

Complexity: $O(n + m)$.

E. Phone Company

Author: Adrian Miclaus
Solved by: 96/99
First to solve: *darkprinceee*

There are $\frac{n(n-1)}{2}$ pairs of students. Each individual selection (choosing one number) can cover at most one pair. Since each of the n students makes K selections, the total number of selections is nK , so necessarily $nK \geq \frac{n(n-1)}{2}$. We can divide by n to get $K \geq \frac{n-1}{2} \iff n \leq 2K+1$. So no construction can work for $n > 2K+1$. Construction achieving $n = 2K+1$: Arrange students on a circle in the order $1, 2, \dots, n$. For each student i , select the next K students: $i+1, i+2, \dots, i+K \pmod{n}$.

F. Language Barrier

Author: Pavle Martinović
Solved by: 4/16
First to solve: VVV

Without loss of generality, we may assume that $R[1] < L[N]$, by mirroring everything if that is not the case. We will solve this problem using dynamic programming, where we calculate the minimum number of moves for person i to obtain the paper in a language he understands. It never makes sense to go to a language with a smaller index, and we can trim all left ends less than $L[1]$ and right ends greater than $R[N]$.

To calculate $dp[u]$ we need to find the minimum of $dp[v] + dist(u, v) + 1$ for all v such that $[L[u], R[u]]$ and $[L[v], R[v]]$ intersect. Additionally, we can reduce the scope to only those where $L[u]$ is contained in $[L[v], R[v]]$, since for other cases where we get to this person using someone with $L[v] > L[u]$ the person who first surpassed the index $L[v]$ could just have translated it to $L[v]$ and sent it directly to v , which would take less turns.

Now, to solve the problem, sort the left and right interval ends into events - $L[u]$ should be used to find the active node with the smallest value of $dp[v] + dist(u, v) + 1$ and then activate v , while $R[v]$ should deactivate v . We're interested in this minimum among all active nodes. This can be done by centroid decomposition - the best path either passes through the centroid or not! In each node of the centroid tree we maintain a set containing the values of $dp[u] + dist(c, u) + 1$ for all u in its (centroid) subtree. Then finding the best path just requires traversing up the centroid tree and taking the value $dist(v, c)$ plus the minimum value of its set. Activating and deactivating is just $\log N$ set insertions/deletions. Calculating distances can be done by precalculating the distances from every node to everything in its centroid subtree, or precalculating LCA and depths. All in all this solves the problem in $O(N \log^2 N)$.

G. Intervals from Triplets

Author: Roman Bilyi
Prepared by: Yarema Stiahar
Solved by: 25/42
First to solve: RAF 1

For each triplet (a_i, b_i, c_i) we must choose two values to form an interval. Notice that the middle value b_i always belongs to any possible interval:

$$[a_i, b_i], [b_i, c_i], [a_i, c_i].$$

Therefore, regardless of the chosen pair, every interval contains b_i .

Since $a_i < b_i < c_i$, if three or more triplets have the same middle value b_i , it is impossible to choose intervals with zero intersection length, because all of them would intersect at b_i . In this case, the answer is -1 .

We sort all triplets by increasing b_i and process them in this order. Triplets with the same value of b_i are processed simultaneously. We use dynamic programming to find the maximum possible total length of intervals.

An interval chosen from the i -th triplet can end either at b_i or at c_i , which is represented by states 0 and 1.

Let $dp_{0,i}$ and $dp_{1,i}$ be the maximum total length using the first i triplets. State 0 means that the chosen interval ends as early as possible, while state 1 means that it ends as late as possible.

More precisely, if the i -th triplet is the only one with value b_i , then:

$$dp_{0,i} \text{ corresponds to an interval ending at } b_i, \quad dp_{1,i} \text{ corresponds to an interval ending at } c_i.$$

If there are two triplets with the same value b_i , then their intervals must end at different right endpoints to avoid intersection. In this case:

$$dp_{0,i} \text{ corresponds to an interval ending at } c_{i-1}, \quad dp_{1,i} \text{ corresponds to an interval ending at } c_i.$$

For each triplet, we try all valid transitions from the previous triplet(s), ensuring that the chosen intervals do not overlap and the total length is maximized. Because at most two triplets have the same middle value, only a constant number of transitions must be considered.

The final answer is the maximum value among all valid DP states after processing all triplets; if no such state exists, the answer is -1 . The overall time complexity is $O(n \log n)$ due to sorting.

H. Sorted Pairs

Author: Roman Bilyi
Prepared by: Yarema Stiahar
Solved by: 0/5
First to solve: N/A

The answer is always at least n , and it is always possible to achieve the goal in $n + 1$ coins.

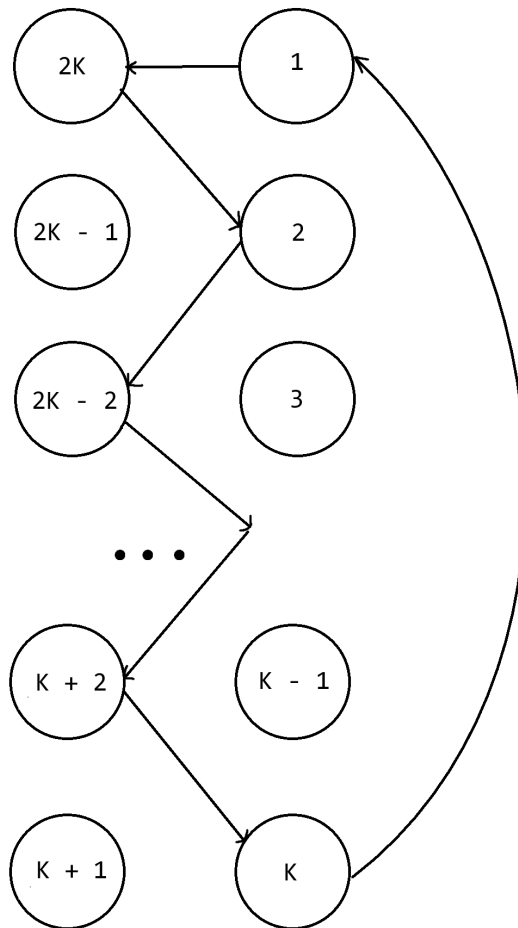
Constructive solution using $n + 1$ coins for any input:

If there are two rows i and j such that

$$a[i][1] < a[j][1] \quad \text{and} \quad a[i][2] > a[j][2],$$

we can fix them with 2 coins by swapping $a[j][1]$ and $a[i][2]$. Repeat this for all such pairs.

After this, for the remaining k rows, a single cycle (see image) can fix all positions using $k + 1$ coins.



Check if n coins is possible:

- Each row must contribute exactly one value to a cycle.
- Each cycle must alternate between first and second column values, so cycles must have even length.

Sort rows by the left value, and let p be the permutation of the right column. We can only move:

- To a larger value: $i \rightarrow j$ if $p[j] > p[i]$.
- To the left: $i \rightarrow j$ if $j < i$.

If some prefix contains all the largest values, it cannot leave that prefix. Thus, the problem can be divided into parts. If any part has odd length, n coins is impossible.

Constructive procedure. For each even-length part:

- Start at the leftmost element, move to the rightmost larger value, then to the smallest left value, repeat until reaching the rightmost element.
- Pair remaining unused elements from left to right, and for each pair:
 - Move left to the smaller element, then move to the larger element.
- Finish the cycle by moving left to the first element.

This constructs the required cycles using exactly n coins whenever all parts have even length.

I. Colorful Components

Author: Alex Popa, Adrian Miclaus
Prepared by: Adrian Miclaus
Solved by: 0/3
First to solve: N/A

Let $G = (V, E)$ be a simple, undirected graph together with a coloring $c : V \rightarrow \mathcal{C}$. A subgraph $G' = (V, E')$ is *feasible* if every connected component of G' is *colorful* (i.e., contains no two vertices with the same color).

Notation. For $X \subseteq V$ let $N(X) = \{u \in V \setminus X : \exists v \in X, (u, v) \in E\}$. For $X \subseteq V$ and $\mathcal{C}_0 \subseteq \mathcal{C}$ let $N_{\mathcal{C}_0}(X) = \{u \in N(X) : c(u) \in \mathcal{C}_0\}$. For a color $\gamma \in \mathcal{C}$ let $V_\gamma = \{v \in V : c(v) = \gamma\}$. We call a tree of diameter 2 a *star*.

Lemma I.1. For each color γ define $s_\gamma = \max_{X \subseteq V_\gamma} (|X| - |N_{\mathcal{C} \setminus \{\gamma\}}(X)|)$. Then any feasible subgraph has at least s_γ singletons of color γ .

Proof. Fix a feasible $G' = (V, E')$ and γ with $s_\gamma > 0$. Let $X \subseteq V_\gamma$ attain the maximum. For every $x \in X$ that is *not* a singleton in G' , pick an arbitrary neighbor $n(x)$ of x inside its (component of) G' . Such $n(x)$ lies in $N_{\mathcal{C} \setminus \{\gamma\}}(X)$, and distinct x lie in distinct components of G' , so the $n(x)$ are pairwise distinct. Hence at most $|N_{\mathcal{C} \setminus \{\gamma\}}(X)|$ vertices of X are non-singletons in G' , implying at least $|X| - |N_{\mathcal{C} \setminus \{\gamma\}}(X)| = s_\gamma$ singletons of color γ . \square

Corollary I.2. Every feasible subgraph has at least $\sum_{\gamma \in \mathcal{C}} s_\gamma$ singletons in total.

We maintain a feasible $G' = (V, E')$, starting from $E' = \emptyset$. As an invariant, every component of G' is always a singleton, an edge, or a star. We process colors $\gamma \in \mathcal{C}$; for a fixed γ let $S_\gamma \subseteq V_\gamma$ be the set of γ -singletons in G' .

An *alternating path* (for γ) in G w.r.t. G' is a simple path whose edges alternate between $E \setminus E'$ and E' , starting at a vertex of S_γ with the first edge in $E \setminus E'$, and ending at a vertex v that satisfies one of:

1. v is a leaf of a star component of G' ; or
2. the component of v in G' contains no vertex of color γ .

Applying such a path p means replacing E' by the symmetric difference $E' \Delta E(p)$.

A BFS from S_γ (expanding via $E \setminus E'$ to non- γ vertices and via E' back to γ vertices) either finds a valid endpoint v and reconstructs p through predecessors, or certifies that none exists.

Lemma I.3. Let p be an alternating path for color γ found by the BFS above, and let \tilde{G} be the result of applying p to G' . Then:

1. the number of γ -singletons drops by one;
2. for every $\gamma' \neq \gamma$, the number of γ' -singletons does not increase;
3. every component of \tilde{G} is colorful; and
4. every component of \tilde{G} is a singleton, an edge, or a star.

Proof. Edges along p are toggled. Because p starts at a γ -singleton and the first edge added connects it to a different color, that start vertex ceases to be a singleton. The endpoint conditions ensure we either detach a leaf from a star or insert a γ vertex into a component without color γ ; in both cases colorful feasibility is preserved and no other color gains new singletons. Local structure along p keeps components within the invariant (singleton/edge/star) family. \square

Lemma I.4. If the BFS for color γ finds no alternating path, then $|S_\gamma| = s_\gamma$.

Proof. The explored search layers implicitly define a maximizing set $X \subseteq V_\gamma$ with $|X| - |N_{C \setminus \{\gamma\}}(X)| = |S_\gamma|$: new neighbors that could enable a decrease are exactly the certified non-existent endpoints. Comparing with Lemma I.1 gives equality. \square

Theorem I.5. *Repeating: for each $\gamma \in \mathcal{C}$, while an alternating path exists, apply it; otherwise move to the next color. The final G' is a feasible subgraph minimizing the number of singletons, and the procedure runs in $O(|V| \cdot |E|)$.*

Proof. By Lemma I.3, each successful application strictly decreases the count of γ -singletons and preserves feasibility and the invariant, hence at most $|V|$ successes occur overall. When the process halts for color γ , Lemma I.4 yields $|S_\gamma| = s_\gamma$; after all colors are processed we have exactly $\sum_\gamma s_\gamma$ singletons, which matches the lower bound of Corollary I.2. \square

J. Triangle Hull

Author: Roman Bilyi
Prepared by: Maksym Shcherba
Solved by: 0/2
First to solve: N/A

If the convex hull of all points is not triangle, the answer is 0. Let A, B, C be the points on convex hull in clockwise direction. Let's check if for the 3 points X, Y, Z all subset convex hulls are triangles.

Firstly, let's add points B and C to the subset, they should be on the convex hull so wlog let the convex hull be X, B, C . If X, Y, Z is good then X, B, C should be good as well. Let's find all possible points X such that triplet X, B, C is good.

Lemma J.1. *Let's sort all points by angle cw around point B starting with point A , sort all points by angle ccw around point C starting with point A . Triplet X, B, C is good iff the prefixes of both orders up to point X are the same.*

Proof. Necessity: If some point D appears only in one prefix (let's say it appears in point B order), the convex hull of X, B, C, D will be B, D, X, C . If two points E, F have different order, the convex hull of X, B, C, E, F will be B, E, F, C .

Sufficiency: All other points are inside triangle XBC so they are irrelevant. If we add some subset of common prefix, the convex hull will be B, C and the point on the common prefix closest to the chain start. □

Let's call the common prefix of those orders A-chain:

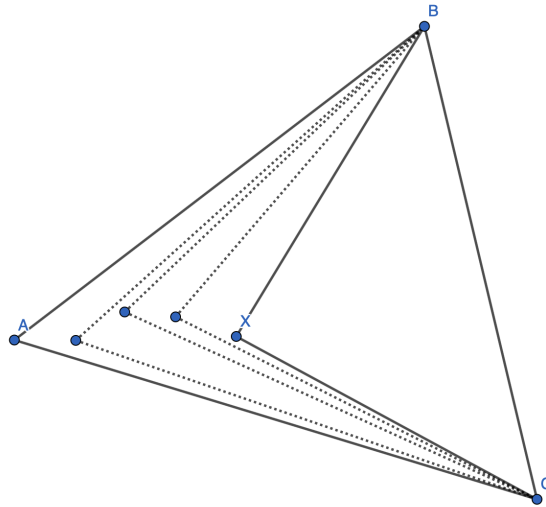


Figure 1: A-chain.

Lemma J.2. *If we add points A, B to the subset X, Y, Z , it's impossible that the same point X remains on the convex hull (X, A, B) .*

Proof. It means that points Y and Z should be inside triangle XBC and triangle XAB simultaneously which is impossible. □

That means that we should have X on A-chain, Y on B-chain, Z on C-chain.

Theorem J.3. *Triplet X, Y, Z is good iff:*

- *The order of all points sorted by angle ccw around point Y starting from YA matches the A-chain up to point X ;*

- The order of all points sorted by angle cw around point Z starting from ZA matches the A-chain up to point X;
- The same for points B-chain and C-chain.

Proof. Necessity: If the order around point Y doesn't match A-chain, we can add point C to the subset, forget about all points outside of triangle AYC and find the same counter example as in Lemma J.1.

Sufficiency:

- Since point Y is on B-chain and point Z is on C-chain there could be no points in red areas.
- Since the sets of points on the prefixes sorted by angle from point Y and point Z are the same, there could be no points in blue areas.
- Points in green area are only points on A-chain and those are in correct order such that it's impossible to add some subset of them to get non-triangle hull.

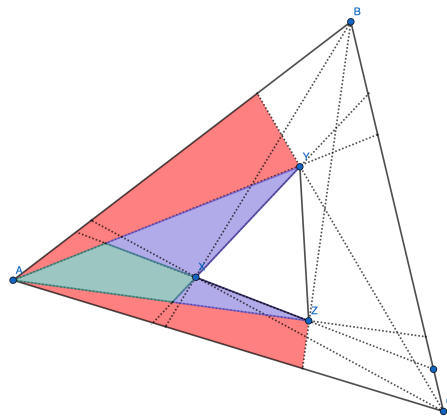


Figure 2: Areas accounted.

If we do the same for B-chain and C-chain, we will cover all area outside of triangle XYZ and only the points on chains are the ones outside. So the convex hull of each subset will be a triangle containing exactly one point on each chain. The point on each chain will be the one that is the closest to the chain start.

□

To calculate the answer we can iterate over all pairs of points (Y, Z), and find the number of possible points X using binary search or 2 pointers.

To summarize we need:

- Compute convex hull, answer is 0 if it's not triangle;
- Find chains by sorting all points around A, B and C.
- For each point, sort all other points around it and compute common prefixes with chains.
- Iterate over Y, Z and find the number of valid points X.

Complexity: $O(n^2 \log n)$.

Bonus. The problem looks doable for $n \leq 10^5$. To compute the answer we can iterate over point Z and maintain segment tree for each Y number of valid X. Geometry part is to check the order is the same look possible using intersection of some half-planes. So is it possible?

K. Connect the Points

Author: Ihor Barenblat

Prepared by: Konstantinos Anemozalis

Solved by: 64/84

First to solve: *UBB_Floricelele_Vesele*

Consider any two pairs of points (A, B) and (C, D) where all 4 points lie on the boundary of the square. If their endpoints appear in the cyclic order A, C, B, D , we consider them to be *interleaved*. Any path connecting A and B within the square, separates C from D , making it impossible to connect them without forming an intersection. Thus, if an *interleaved* pair exists, the answer is NO.

Otherwise, it's always possible to construct a solution. Let's classify the pairs into three types:

- Type A: Pairs with exactly one point on the boundary.
- Type B: Pairs with both points on the boundary.
- Type C: Pairs with neither point on the boundary.

Intuitively, since Type B pairs are not *interleaved*, they can be connected using curves that stay arbitrarily close to the boundary, nesting inside one another where necessary. These paths partition the square into several regions, but never in a way that separates the two endpoints of any remaining pair.

The construction is as follows:

1. First, connect pairs of Type A. A curve from a boundary point to an internal point does not disconnect the square, so the remaining free space stays connected; thus any two points that still need to be connected can be joined by a curve avoiding the already drawn ones.
2. Next, let's consider pairs of Type B. Since they are not *interleaved*, we can route curves close to the boundary, going around the curves created by Type A connections and nesting around other Type B paths. Each such curve separates the square into two regions, and by the non-interleaving property, every remaining Type B pair has both endpoints in exactly one of these regions.
3. Finally, pairs of Type C lie entirely in the remaining connected interior. We can connect an arbitrary pair with a curve and the region remains connected, thus all such points can be joined.

A simple $O(N^2)$ solution that checks for the *interleaved* condition is good enough, but the problem can also be solved in $O(N)$ by ordering boundary points and using a stack (similarly to checking a valid parenthesis sequence).

L. Neo-Nim

Author: Ihor Barenblat, Vladyslav Zavodnyk
Prepared by: Konstantinos Anemozalis
Solved by: 5/21
First to solve: *Sigma--1*

First, we observe two sufficient conditions for Bob to win from a given state:

1. If there is any pile with $a_i = 1$, Bob can simply save this pile for the end. Ana must eventually remove ≥ 2 stones from other piles or run out of moves, leaving Bob to remove the 1.
2. If there are two or more piles with $a_i = 2$, Bob wins. Ana is forced to clear one of them ($2 \rightarrow 0$), then Bob removes 1 stone from the other ($2 \rightarrow 1$), creating a pile of 1 (reverting to the previous winning case).

Thus, for Ana to have any chance of winning, the initial array must contain no 1s and at most one 2. Now let's consider the cases for each k :

Case $k = 2$

In this variation, every pair of moves reduces the total stone sum by 3.

- If there exists a pile with $a_i \equiv 1 \pmod{3}$, Bob wins. If Ana touches it, Bob can also move on it to maintain the invariant; eventually, the pile will be reduced to 1 stone.
- If there are two piles with $a_i \equiv 2 \pmod{3}$, Bob wins. He can reduce one of them to 1 ($\pmod{3}$) on his turn, forcing the previous condition.

Ana wins if and only if she can make the last move. This requires the initial total sum $S \equiv 2 \pmod{3}$, and the configuration must not allow Bob to force a win. Therefore, Ana wins if and only if there is exactly one pile with $a_i \equiv 2 \pmod{3}$ and all other piles satisfy $a_i \equiv 0 \pmod{3}$.

Case $k = 3$

We classify the pile sizes based on their strategic advantage:

- **3:** If Ana sees a 3, she removes it ($3 \rightarrow 0$), effectively passing the turn while removing a pile. If Bob sees a 3, he plays $3 \rightarrow 2$, forcing Ana to respond with $2 \rightarrow 0$. The pile is removed, and it remains Bob's turn (neutral for Bob).
- **4:** Neutral, similar to 0. Ana cannot touch a 4 (leaves 1 or 2, leading to a loss), so she must wait for Bob to reduce it to 3, then she clears it.
- **5:** Acts as a 1 move advantage to the player who uses it:
 - If Ana uses it: $(A : 5 - 2 = 3) \rightarrow (B : 3 - 1 = 2) \rightarrow (A : 2 - 2 = 0) \rightarrow$ Bob's turn.
 - If Bob uses it: $(B : 5 - 1 = 4) \rightarrow 4$ (neutral) \rightarrow Ana's turn.
 - Note that two 5s are neutral as they cancel each other out.
- **6:** Advantage for Ana. She plays $6 \rightarrow 3$, creating a pile of 3.
- ≥ 7 : Winning for Ana. She can always manipulate these large piles to gain a move advantage.

Subcase 1: No pile of 2 stones

Ana moves first. She wins if she can claim an advantage, i.e., if any of the following are true:

- There is at least one pile of 3.
- The count of piles with 5 stones is odd.

- There is any number ≥ 6 (she converts $6 \rightarrow 3$ or $7 \rightarrow 4$, etc.).

Otherwise, the board is neutral (i.e., only 4s and an even number of 5s), and Bob wins.

Subcase 2: Exactly one pile of 2 stones

Ana is forced to play $2 \rightarrow 0$ immediately to prevent Bob from creating a 1. It is now Bob's turn to move on the remaining piles. Any existing 3s are neutralized by Bob, so they provide no advantage to Ana. Bob wins if and only if he has a strict advantage in the remaining configuration:

1. The count of piles with 5 stones is odd (giving Bob the parity advantage).
2. AND there is at most one pile of 6 stones (if there are two 6s, Ana can use one to regain the advantage).
3. AND there are no piles ≥ 7 .

Case $k \geq 4$

If the initial Bob winning conditions (a pile of 1 or ≥ 2 piles of 2) are not met, Ana wins. Since $k \geq 4$, Ana has enough freedom to choose x such that she maintains the invariant that all $a_i \geq 3$ after her turn, or she can simply mimic the winning strategies from $k = 3$ with even more flexibility.