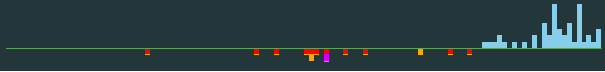
B: Balloons and Dubloons

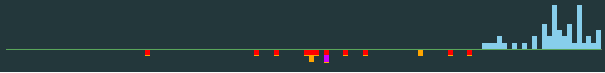
Problem author: Marijn Adriaanse



Problem: Given a treasure map with fake markers, add a single marker to communicate the treasure.

B: Balloons and Dubloons

Problem author: Marijn Adriaanse

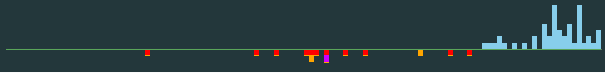


Problem: Given a treasure map with fake markers, add a single marker to communicate the treasure.

Observation: The water does not matter, neither does the map. We can flatten the problem into $n + k$ land tiles in a row, with some marked.

B: Balloons and Dubloons

Problem author: Marijn Adriaanse



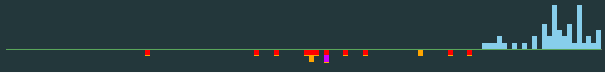
Problem: Given a treasure map with fake markers, add a single marker to communicate the treasure.

Observation: The water does not matter, neither does the map. We can flatten the problem into $n + k$ land tiles in a row, with some marked.

Equivalent: Find a hash function mapping a set of $k + 1$ markers to the treasure, used by the matey. The hash should allow the captain to always add a working marker.

B: Balloons and Dubloons

Problem author: Marijn Adriaanse



Problem: Given a treasure map with fake markers, add a single marker to communicate the treasure.

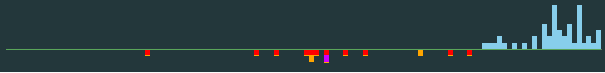
Observation: The water does not matter, neither does the map. We can flatten the problem into $n + k$ land tiles in a row, with some marked.

Equivalent: Find a hash function mapping a set of $k + 1$ markers to the treasure, used by the matey. The hash should allow the captain to always add a working marker.

Solution: Take the sum of all marker indices, modulo $k + 1$, as the hash.

B: Balloons and Dubloons

Problem author: Marijn Adriaanse



Problem: Given a treasure map with fake markers, add a single marker to communicate the treasure.

Observation: The water does not matter, neither does the map. We can flatten the problem into $n + k$ land tiles in a row, with some marked.

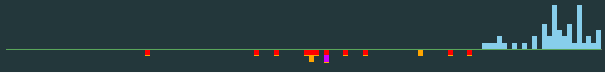
Equivalent: Find a hash function mapping a set of $k + 1$ markers to the treasure, used by the matey. The hash should allow the captain to always add a working marker.

Solution: Take the sum of all marker indices, modulo $k + 1$, as the hash.

Pitfall: Do not recompute the hash for every marker location. This is too slow.

B: Balloons and Dubloons

Problem author: Marijn Adriaanse



Problem: Given a treasure map with fake markers, add a single marker to communicate the treasure.

Observation: The water does not matter, neither does the map. We can flatten the problem into $n + k$ land tiles in a row, with some marked.

Equivalent: Find a hash function mapping a set of $k + 1$ markers to the treasure, used by the matey. The hash should allow the captain to always add a working marker.

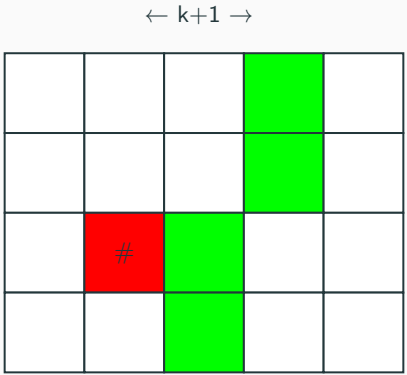
Solution: Take the sum of all marker indices, modulo $k + 1$, as the hash.

Pitfall: Do not recompute the hash for every marker location. This is too slow.

Pitfall: Taking the marker index from left to right does not always work! A fix is to index from right to left instead.

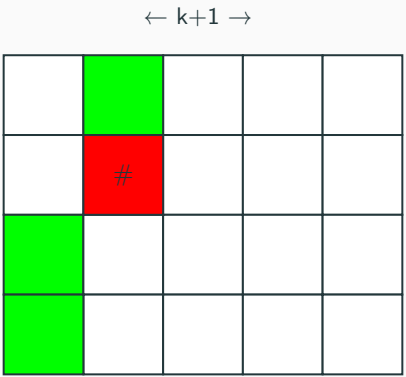
B: Balloons and Dubloons

Problem author: Marijn Adriaanse



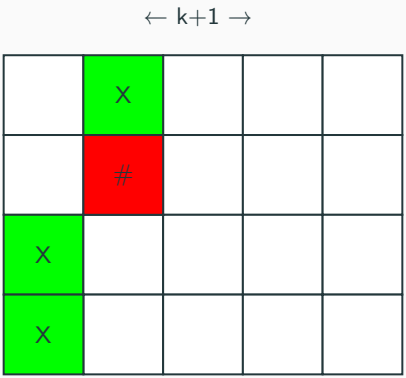
B: Balloons and Dubloons

Problem author: Marijn Adriaanse



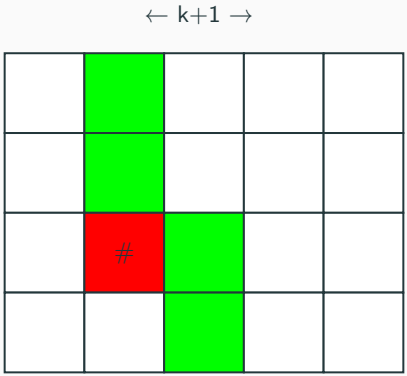
B: Balloons and Dubloons

Problem author: Marijn Adriaanse



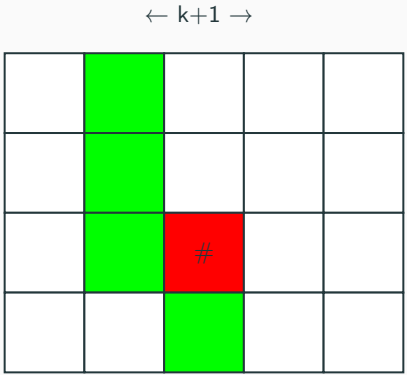
B: Balloons and Dubloons

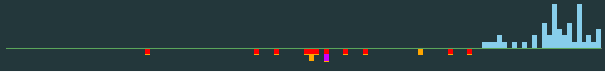
Problem author: Marijn Adriaanse



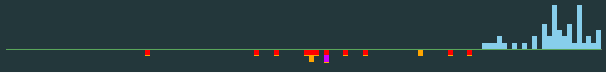
B: Balloons and Dubloons

Problem author: Marijn Adriaanse





Cheese: Your hash does not need to be perfect. You can hardcode small cases, and randomly permute the maps on larger inputs to pass all tests.

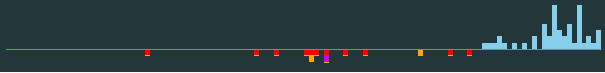


Cheese: Your hash does not need to be perfect. You can hardcode small cases, and randomly permute the maps on larger inputs to pass all tests.

Running time: $\mathcal{O}(n)$.

B: Balloons and Dubloons

Problem author: Marijn Adriaanse



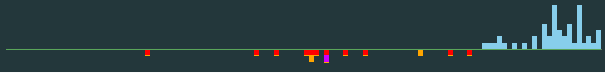
Cheese: Your hash does not need to be perfect. You can hardcode small cases, and randomly permute the maps on larger inputs to pass all tests.

Running time: $\mathcal{O}(n)$.

Fun Fact: The bound $n > k^2$ can be lowered to $n \geq \Omega(k \log k)$, using the Lovász local lemma (the other LLL).

B: Balloons and Dubloons

Problem author: Marijn Adriaanse



Cheese: Your hash does not need to be perfect. You can hardcode small cases, and randomly permute the maps on larger inputs to pass all tests.

Running time: $\mathcal{O}(n)$.

Fun Fact: The bound $n > k^2$ can be lowered to $n \geq \Omega(k \log k)$, using the Lovász local lemma (the other LLL).

Statistics: 61 submissions, 0 accepted, 47 unknown