D – Run Lads Run

Luo Tsz Fung {pepper1208}

2025-05-16

Background

Problem Idea by `pepper1208, USACO`

Preparation by `pepper1208, rina__owo`

Problem Restatement

Given *N* blocks consecutively. Each block is unclassified, or classified as type A or B.

A runner runs through the *N* blocks. If the runner runs through two adjacent blocks with the same type, it will experience a speed boost.

You can determine the type of all unclassified blocks. Output all possible number of speed boost the runner receives with any determination of types of all unclassified blocks.

Subtasks Constraints

	Score	Constraints
1	28	$N \leq 10$
2	72	No additional constraints

Ad-hoc Problem

- Ad-hoc problems are problems that do not fall into standard categories with well-studied solutions.
- Each ad-hoc problem is different, and no specific or general techniques exist to solve them.
- The only thing you can rely on when solving ad-hoc problem is your *brain power*, with a lot of drafting and observations.

How to tackle ad-hoc problem?

- Draw lots of small cases to gain better understanding of the problem. It can guide you to start with the problem, or help you debug your code, or guide you to further approach the problem.
- Whenever you find an observation that seems useful, write it down.
- Do not get stuck on any specific idea, unless you see an entire solution.
- Try to approach the problem from a lot of different perspectives with visual depiction.
- Do a lot of ad-hoc problems.

Before full solution: Subtask 1

- Brute-force generate every possible string *S*.
- Time complexity: $O(N * 2^N)$
- Expected score: **28**

Observations

- The way to achieve full solution is observing different simple cases.
- Also, it is recommended to actually understand *what the problem wants you to find*.

Observations

- Consider the pattern AX…XA. What is the output of it?
- AA: 1
- AXA: 0, 2
- AXXA: 1, 3
- AXXXA: 0, 2, 4
- AXXXXA: 1, 3, 5
- …

Observations

- We can conclude that: For the pattern AX…XA,
- If N is even, the solution is 1, 3, 5, …, $N - 1$.
- If N is odd, the solution is 0, 2, 4, …, $N - 1$.
- The proof is left as exercise. ☺

Observations

- Nevertheless, for the pattern AX…XB,
- If N is even, the solution is 0, 2, 4, …, $N – 2$.
- If N is odd, the solution is 1, 3, 5, …, $N – 2$.
- The proof is left as exercise. ☺

Observations

- Consider the pattern AX…XABX…XB. What is the output of it?
- It seems that we can decompose the pattern into two partitions: AX…XA and BX…XB.
- Find the possible number for both patterns, and add them up.
- Note that for the pattern AX…XAAX…XA, we need to add 1 to each of the answers since the middle two *A*'s contribute 1 to the answer.

Observations

- Consider the pattern X…XA. What is the output of it?
- If the pattern length is *N*, we can get any value 0, 1, 2, 3, …, *N* – 1.

Observations

- Consider the pattern ABXABXABX…. What is the output of it?
- If ABX is repeated *N* times, our answer is *N* – 1 and *N*.

Observations

- Consider the pattern AAA…. What is the output of it?
- If the pattern length is *N*, the answer is *N* – 1.

Observations

- Consider the pattern XXX…. What is the output of it?
- If the pattern length is *N*, the answer is 0, 1, …, *N* – 1.

Full solution

- Now, we are going to come up with a solution for the general problem using the observations we came up with earlier.
- Assume the starting / ending is not *X*.
- Split the string into "sections" such that each section has one starting character that isn’t *X*, one ending character that isn’t *X*, and a bunch of *X*'s in the middle. The edge characters of each "section" can overlap.

Full solution

- For starting and ending of a section is the same,
  - If N is even, min is 1 and max is $N - 1$. Otherwise, min is 0 and max is $N - 1$.
- For starting and ending of a section is not the same,
  - If N is even, min is 0 and max is $N - 2$. Otherwise, min is 0 and max is $N - 2$.
- For any min x and max y, we can achieve any value x, $x + 2$, $x + 4$, …, $y - 2$, y.
- We can add the min / max values from every section together to get the min / max values for the whole string.

Full solution

- Now, let *x* be the number of *X*'s at the start of the pattern and *y* be the number of *X*'s at the end of the pattern.
- These *X* contributes a minimum of 0 and a maximum of *x* + *y*.
- If *x* is not 0 or *y* is not 0, we can achieve all numbers for the pattern of every value in between min and max.
- Otherwise, we can still achieve numbers for the pattern with the increment of 2.
- Last but not least, handle the case where the entire pattern is *X*.

Full solution

- With careful implementation, you can solve this problem!
- Time complexity: O(N)
- Expected score: **100 AC!**

Takeaways

- The first step is solving the sample inputs by hand. The goal is to make sure you fully understand what the problem is asking you to do.
- We might want to test a lot of simple test cases that have some pattern to them. We want to use these to uncover patterns that we might be able to use to solve the problem.
- For every pattern, try to come up with a possible solution, as well as a justification as to why that solution is correct. Not all patterns will have nice solutions. If you can't think of anything, just move on.