Competitive Programming and Mathematics Society

Programming Workshop #1

Progressive Problem Solving

Patrick Moore and Ryan Ong

Today's Workshop

CPMSOC

1. Subtasks
2. Problem: Combo
3. Problem: Gameboy
4. Problem: Martian DNA
5. Wrap up

Subtasks: An important problem solving tool

CPMSOC

Theorem

Subtasks are designed to help you solve the problem. Problem writers will often write subtasks not just to reward partial algorithms, but also to lead you towards the full solution.

It can be very useful to try and identify the most interesting subtasks. These might be ones that impose strange constraint on the problem, especially when working with multi-variable problems.

We will try and get some practice with this in the problems today.

Problem: Combo

CPMSOC

There is a secret string, S, of length N which consists of four characters: 'A', 'B', 'X' or 'Y'. You know that the first character of S does not appear in the rest of S.

You can ask a query of length up to $4N$, and you get the size of the largest prefix of S that is in your query string.

Full: Use at most $N + 2$ queries

Problem: Combo

CPMSOC

There is a secret string, S, of length N which consists of four characters: 'A', 'B', 'X' or 'Y'. You know that the first character of S does not appear in the rest of S.

You can ask a query of length up to $4N$, and you get the size of the largest prefix of S that is in your query string.

Full: Use at most $N + 2$ queries

- $Q = 4N$

Problem: Combo

CPMSOC

There is a secret string, S, of length N which consists of four characters: 'A', 'B', 'X' or 'Y'. You know that the first character of S does not appear in the rest of S.

You can ask a query of length up to $4N$, and you get the size of the largest prefix of S that is in your query string.

Full: Use at most $N + 2$ queries

- Q = $4N$ - Hint: Brute Force?

Problem: Combo

CPMSOC

There is a secret string, S, of length N which consists of four characters: 'A', 'B', 'X' or 'Y'. You know that the first character of S does not appear in the rest of S.

You can ask a query of length up to $4N$, and you get the size of the largest prefix of S that is in your query string.

Full: Use at most $N + 2$ queries

- Q = $4N$ - Hint: Brute Force?
- Q = $2N + 1$

Problem: Combo

CPMSOC

There is a secret string, S, of length N which consists of four characters: 'A', 'B', 'X' or 'Y'. You know that the first character of S does not appear in the rest of S.

You can ask a query of length up to $4N$, and you get the size of the largest prefix of S that is in your query string.

Full: Use at most $N + 2$ queries

- Q = $4N$ - Hint: Brute Force?
- Q = $2N + 1$ - Hint: Can we get 2 queries per item of S?
- Q = $N + 2$

Problem: Combo

CPMSOC

There is a secret string, S, of length N which consists of four characters: 'A', 'B', 'X' or 'Y'. You know that the first character of S does not appear in the rest of S.

You can ask a query of length up to $4N$, and you get the size of the largest prefix of S that is in your query string.

Full: Use at most $N + 2$ queries

- Q = $4N$ - Hint: Brute Force?
- Q = $2N + 1$ - Hint: Can we get 2 queries per item of S?
- Q = $N + 2$ - Hint: Can we exploit the query size of $4N$ to get down to 1 query per item of S?

Problem: Gameboy

CPMSOC

Remember that T = maxWrites + maxReads; full is for T = 10

Problem: Gameboy

CPMSOC

Remember that T = maxWrites + maxReads; full is for T = 10
How can we minimise writes?
How can we minimise reads?

Problem: Gameboy

CPMSOC

Remember that T = maxWrites + maxReads; full is for T = 10
How can we minimise writes?
How can we minimise reads?

- T = 24

Problem: Gameboy

CPMSOC

Remember that T = maxWrites + maxReads; full is for T = 10
How can we minimise writes?
How can we minimise reads?

- T = 24 - Hint: think binary

Problem: Gameboy

CPMSOC

Remember that T = maxWrites + maxReads; full is for T = 10
How can we minimise writes?
How can we minimise reads?

- T = 24 - Hint: think binary
- T = 17

Problem: Gameboy

CPMSOC

Remember that T = maxWrites + maxReads; full is for T = 10
How can we minimise writes?
How can we minimise reads?

- T = 24 - Hint: think binary
- T = 17 - Hint: think tertiary

Problem: Gameboy

CPMSOC

Remember that T = maxWrites + maxReads; full is for T = 10
How can we minimise writes?
How can we minimise reads?

- T = 24 - Hint: think binary
- T = 17 - Hint: think tertiary
- T = 14

Problem: Gameboy

CPMSOC

Remember that T = maxWrites + maxReads; full is for T = 10
How can we minimise writes?
How can we minimise reads?

- T = 24 - Hint: think binary
- T = 17 - Hint: think tertiary
- T = 14 - Hint: think quaternary

Problem: Gameboy

CPMSOC

Remember that T = maxWrites + maxReads; full is for T = 10
How can we minimise writes?
How can we minimise reads?

- T = 24 - Hint: think binary
- T = 17 - Hint: think tertiary
- T = 14 - Hint: think quaternary
- T = 12

Problem: Gameboy

CPMSOC

Remember that T = maxWrites + maxReads; full is for T = 10
How can we minimise writes?
How can we minimise reads?

- T = 24 - Hint: think binary
- T = 17 - Hint: think tertiary
- T = 14 - Hint: think quaternary
- T = 12 - Hint: Can we get a slight optimisation on quaternary?

Problem: Gameboy

CPMSOC

Remember that T = maxWrites + maxReads; full is for T = 10
How can we minimise writes?
How can we minimise reads?

- T = 24 - Hint: think binary
- T = 17 - Hint: think tertiary
- T = 14 - Hint: think quaternary
- T = 12 - Hint: Can we get a slight optimisation on quaternary?
- T = 11

Problem: Gameboy

CPMSOC

Remember that T = maxWrites + maxReads; full is for T = 10
How can we minimise writes?
How can we minimise reads?

- T = 24 - Hint: think binary
- T = 17 - Hint: think tertiary
- T = 14 - Hint: think quaternary
- T = 12 - Hint: Can we get a slight optimisation on quaternary?
- T = 11 - Hint: Can we optimise the base?

Problem: Gameboy

CPMSOC

Remember that T = maxWrites + maxReads; full is for T = 10
How can we minimise writes?
How can we minimise reads?

- T = 24 - Hint: think binary
- T = 17 - Hint: think tertiary
- T = 14 - Hint: think quaternary
- T = 12 - Hint: Can we get a slight optimisation on quaternary?
- T = 11 - Hint: Can we optimise the base?
- T = 10

Problem: Gameboy

CPMSOC

Remember that T = maxWrites + maxReads; full is for T = 10
How can we minimise writes?
How can we minimise reads?

- T = 24 - Hint: think binary
- T = 17 - Hint: think tertiary
- T = 14 - Hint: think quaternary
- T = 12 - Hint: Can we get a slight optimisation on quaternary?
- T = 11 - Hint: Can we optimise the base?
- T = 10
  Hint: Do we have to have a consistent base the whole time?

Problem: Gameboy

CPMSOC

Remember that T = maxWrites + maxReads; full is for T = 10
How can we minimise writes?
How can we minimise reads?

- T = 24 - Hint: think binary
- T = 17 - Hint: think tertiary
- T = 14 - Hint: think quaternary
- T = 12 - Hint: Can we get a slight optimisation on quaternary?
- T = 11 - Hint: Can we optimise the base?
- T = 10
  Hint: Do we have to have a consistent base the whole time?
  Hint: Can we decide on a number of layers first? Then determine what base should be used each time?

Problem: Martian DNA

CPMSOC

There is a secret binary string S, of length n. You can make queries of the form:

- Is P a substring of S?

Guess this secret string in the least queries.

Subtasks:

- $n \leq 5, t = 31$
- $n \leq 100, t = 256$
- $n \leq 1000, t = 1024$

NOTE: t is the most number of queries allowed.
NOTE: The grader is adaptive, which means that a randomised solution wont work :'(

Problem: Martian DNA

CPMSOC

e.g. The secret string is "101"
query("00") -> false.
query("01") returns true.
query("10") returns true.
query("11") returns false.
query("101") returns true.

https://oj.uz/problem/view/IOI16_dna

Problem: Martian DNA

CPMSOC

Subtask 1

$n \leq 5, t = 31$.

- $31 = 2^5 - 1$
- Which seems to imply a brute force...

Problem: Martian DNA

CPMSOC

Subtask 2

$n \leq 100, t = 256$

- $t \geq 2n$
- Helpful to know the ends of the string

Problem: Martian DNA

CPMSOC

Subtask 3

$n \leq 1000, t = 1024$

- Going for $N + 2\log_2 N$ queries
- look for long sequence of 1's

Wrap up

CPMSOC

- Problems:
  - IOI 2010 Memory [Easy] (https://ioi2010.org/Tasks/Day2/Memory.shtml)
  - Freq Fish [Medium] (https://pastebin.com/inUavLsb)
  - CEOI 2014 Question [Hard] (http://ceoi.inf.elte.hu/probarch/14/question.pdf)
  - Worm Worries [Very Hard] (https://boi18-day1-open.kattis.com/problems/worm)
- Next competitive programming workshop is in week 8.
- A reminder about the competitive maths workshops that run on Wednesdays, 12-2 in Week 7 and 9.