Hong Kong Olympiad in Informatics 2007 Senior Question 3

> Hackson Leung

Statistics

Problem Description

ivalve Solution

50% Solution

70% Solution

100% Solutio

Special Note

Hong Kong Olympiad in Informatics 2007 Senior Question 3 SOS

Hackson Leung

13th January, 2007

Hong Kong Olympiad in Informatics 2007 Senior Question 3

> Hackson Leung

Statistics

Problem Description

Naïve Solutio

E00/ C.L.

70% Solution

100% Solutio

Special Note

${\sf Disappointing}$

Hong Kong Olympiad in Informatics 2007 Senior Question 3

> Hackson Leung

Statistics

Problem
Description

Naïve Soluti

50% Solution

70% Solution

Special Not

Disappointing

HKOI 2007 Senior Q3 Statistics

Hong Kong Olympiad in Informatics 2007 Senior Question 3

> Hackson Leung

Statistics

Problem
Description

Naïve Solut

50% Solution

70% Solution

Disappointing

HKOI 2007 Senior Q3 Statistics

Max: 100

Hong Kong Olympiad in Informatics 2007 Senior Question 3

> Hackson Leung

Statistics

Problem Description

50% Solution

70% Solution

100% Solution

Special Not

Disappointing

HKOI 2007 Senior Q3 Statistics

Max : 100

#Max : 3

Hong Kong Olympiad in Informatics 2007 Senior Question 3

> Hackson Leung

Statistics

Problem Description

.....

50% Solution

70% Solution

C

Disappointing

HKOI 2007 Senior Q3 Statistics

Max : 100

#Max : 3

 $\mathsf{Min}:\, \mathsf{0}$

Hong Kong Olympiad in Informatics 2007 Senior Question 3 SOS

> Hackson Leung

Statistics

Problem Description

.

3070 3010101

70% Solution

Special Not

Disappointing

HKOI 2007 Senior Q3 Statistics

Max : 100

#Max : 3

Min : 0

Mean: 28.627

Hong Kong Olympiad in Informatics 2007 Senior Question 3

> Hackson Leung

Statistics

Problem Description

Naive Solution

700/ Caladia

70% Solution

Special Not

Disappointing

HKOI 2007 Senior Q3 Statistics

Max : 100

#Max : 3

Min: 0

Mean: 28.627

Mode : 10

Hong Kong Olympiad in Informatics 2007 Senior Question 3 SOS

> Hackson Leung

Statistics

Problem Description

ivalve Solution

70% Solutio

1070 30141101

Special Note

Disappointing

HKOI 2007 Senior Q3 Statistics

Max: 100

#Max : 3

Min : 0

Mean: 28.627

Mode: 10

Std. Dev.: 30.423

Hong Kong Olympiad in Informatics 2007 Senior Question 3 SOS

> Hackson Leung

Statistics

Problem Description

Naïve Solution

.

Special Note

Disappointing

HKOI 2007 Senior Q3 Statistics

Max : 100

#Max : 3

Min: 0

Mean: 28.627

Mode: 10

Std. Dev.: 30.423

Or7

Hong Kong Olympiad in Informatics 2007 Senior Question 3

> Hacksor Leung

Statistics

Problem Description

Naïve Salutio

0% Solution

70% Solution

100% Solutio

Hong Kong Olympiad in Informatics 2007 Senior Question 3

> Hackson Leung

Statistics

Problem

Description

TValve Solution

70% 3010110

Special Not

• Given a $N \times M$ characters map, each character can be a brick or a space

Hong Kong Olympiad in Informatics 2007 Senior Question 3

> Hacksor Leung

Statistic

Problem Description

Naïve Solution

50% Solution

70% Solution

100/0 501411

■ Given a $N \times M$ characters map, each character can be a brick or a space

You can rotate each row to right or to left

Hong Kong Olympiad in Informatics 2007 Senior Question 3

> Hackson Leung

Statistic

Problem Description

Naïve Solution

700/ Calusia

70% 3010110

- Given a $N \times M$ characters map, each character can be a brick or a space
- You can rotate each row to right or to left
- Find the minimum number of rotation such that there exists at least one column of *N* spaces

Hong Kong Olympiad in Informatics 2007 Senior Question 3

> Hackson Leung

Statistics

Problem

Naïve Solution

50% Solution

70% Solution

100% Solutio

Hong Kong Olympiad in Informatics 2007 Senior Question 3

> Hackson Leung

Statistics

Problem
Description

Naïve Solution

70% Salutio

Caralal Nati

Asking for combinations

Hong Kong Olympiad in Informatics 2007 Senior Question 3

> Hackson Leung

Statistic

Problem
Description

Naïve Solution

70% Salutio

10/0 3010110

C. I NI. .

- Asking for combinations
- Asking for optimal solution

Hong Kong Olympiad in Informatics 2007 Senior Question 3

> Hackson Leung

Statistic

Problem Description

Naïve Solution

50% Solution

70% Solutio

1000/ 6-1-2

- Asking for combinations
- Asking for optimal solution
- Exhuastion!?

Naïve Implementation

Hong Kong Olympiad in Informatics 2007 Senior Question 3

> Hacksor Leung

Statistics

Problem Description

Naïve Solution

50% Solution

70% Solution

100% Solutio

Naïve Implementation

```
Hong Kong
Olympiad in
Informatics
2007
Senior
Question 3
SOS
```

Hacksor Leung

Statistic

Problem Description

Naïve Solution

70% Salution

70% Solution

```
Naïve Implementation
Algorithm exhaustion(ROW, STEP)
   Mark this configuration as visited
   if ROW = N and At least one column has N spaces then
     Answer = Min(Answer, STEP)
   else
     While True do
       Rotate ROW to left
       if current configuration is not visited then
         Exhaustion(ROW+1, STEP+Number_of_Left_Rotation)
       else
         break
     Return to original configuration
     {Do the same for Right Rotation}
```

Hong Kong Olympiad in Informatics 2007 Senior Question 3

> Hacksor Leung

Statistics

Problem

Naïve Solution

0% Solutio

70% Solution

100% Solutio

Hong Kong Olympiad in Informatics 2007 Senior Question 3

> Hackson Leung

Statistics

Problem
Description

Naïve Solution

E00/ C 1 .:

70% Solution

100% Saluti

Special Note

Clumsy

Hong Kong Olympiad in Informatics 2007 Senior Question 3

> Hackson Leung

Statistics

Problem
Description

Naïve Solution

50% Solution

70% Solution

100% Soluti

- Clumsy
- Risk of TLE

Hong Kong Olympiad in Informatics 2007 Senior Question 3

> Hackson Leung

Statistics

Problem
Description

Naïve Solution

70% Solutio

Caralal Nas

- Clumsy
- Risk of TLE
- When you were implementing this, probably you can think of 50% Solution

Hong Kong Olympiad in Informatics 2007 Senior Question 3

> Hacksor Leung

Statistics

Problem Description

Naïve Solutio

50% Solution

70% Solution

Hong Kong Olympiad in Informatics 2007 Senior Question 3

> Hacksor Leung

Statistics

Problem
Description

Naïve Solutio

50% Solution

70% Solutio

Special Note

1 No need to search for all possible combinations

Hong Kong Olympiad in Informatics 2007 Senior Question 3

> Hacksor Leung

Statistics

Problem
Description

Naïve Solutio

50% Solution

70% Solutio

100% Solution

- 1 No need to search for all possible combinations
- 2 One row can have at most M different configurations

Hong Kong Olympiad in Informatics 2007 Senior Question 3

> Hacksor Leung

Statistic

Problem Description

50% Solution

70% Solutio

Caralal Nas

- 1 No need to search for all possible combinations
- 2 One row can have at most M different configurations
- 3 For each character, we should know the number of minimum rotation made to "pull" a space to that position

Hong Kong Olympiad in Informatics 2007 Senior Question 3 SOS

> Hacksor Leung

Statistics

Problem Description

Naïve Solution
50% Solution

70% Solutio

. 0,0 00.00.0

- 1 No need to search for all possible combinations
- 2 One row can have at most M different configurations
- 3 For each character, we should know the number of minimum rotation made to "pull" a space to that position
- 4 Then we can find the minimum roration by comparing the sum

Hong Kong Olympiad in Informatics 2007 Senior Question 3 SOS

> Hackson Leung

Statistics

Problem Description

Naïve Solution
50% Solution

70% Solution

100/0 3010110

- 1 No need to search for all possible combinations
- 2 One row can have at most M different configurations
- For each character, we should know the number of minimum rotation made to "pull" a space to that position
- 4 Then we can find the minimum roration by comparing the sum
- 5 If one level is rotated to left K times, it is equivalent to rotate to right $(M K) \mod M$ times

50% Solution - Implementation

Hong Kong Olympiad in Informatics 2007 Senior Question 3

> Hackson Leung

Statistics

Problem Description

Naïve Solutio

50% Solution

70% Solution

1000/ 5-1....:-

50% Solution - Implementation

Hong Kong Olympiad in Informatics 2007 Senior Question 3 SOS

> Hacksor Leung

Statistic

Problem Description

Naïve Solution

70% Solutio

70% Solution

Special Note

50% Solution

Algorithm 50% Solution MinRotation[][] $\leftarrow \infty$

MinRotation[x][y] $\leftarrow 0$ $//\exists x, y \rightarrow (x, y)$ is a space

for each character (r, c) do

if (r,c) is a space then

for $i \leftarrow 1$ to M-1 do

Update MinRotation at (r, c) rotates to left i times with i Update MinRotation at (r, c) rotates to right i times with i

$$Answer \leftarrow \min_{1 \leq i \leq M} \{ \sum_{j=1}^{N} MinRotation[j][i] \}$$

Hong Kong Olympiad in Informatics 2007 Senior Question 3

> Hacksor Leung

Statistics

Problem Description

Naïve Solutio

50% Solution

70% Solution

1000/ 6-1-4

Hong Kong Olympiad in Informatics 2007 Senior Question 3 SOS

> Hackson Leung

Statistics

Problem Description

Naïve Solution

50% Solution

70% Salutio

10/0 3010110

Easy to implement

Hong Kong Olympiad in Informatics 2007 Senior Question 3

> Hackson Leung

Statistics

Problem
Description

Naïve Solution

50% Solution

70% Salutio

70/0 3010110

- Easy to implement
- Reduce memory usage

Hong Kong Olympiad in Informatics 2007 Senior Question 3

> Hackson Leung

Statistic

Problem
Description

Naïve Solutio

50% Solution

70% Solutio

100% Solutio

- Easy to implement
- Reduce memory usage
- But still, it is not efficient enough to get 100% score

Hong Kong Olympiad in Informatics 2007 Senior Question 3

> Hackson Leung

Statistic

Problem
Description

Naive Solution

50% Solution

70% Solutio

- Easy to implement
- Reduce memory usage
- But still, it is not efficient enough to get 100% score
- Runtime Complexity: $\mathcal{O}(NM^2)$

Hong Kong Olympiad in Informatics 2007 Senior Question 3

> Hackson Leung

Statistics

Problem Description

Naïve Solutio

50% Solutio

70% Solution

100% 3010110

Hong Kong Olympiad in Informatics 2007 Senior Question 3 SOS

> Hackson Leung

Statistics

Problem
Description

Naïve Solution

50% Solution

70% Solution

1000/ 6-1-4

Special Note

■ The bottleneck is "How to find the nearest space to an element" efficiently

Hong Kong Olympiad in Informatics 2007 Senior Question 3 SOS

> Hacksor Leung

Statistic

Problem
Description

Naïve Solutio

50% Solution

70% Solution

- The bottleneck is "How to find the nearest space to an element" efficiently
- Consider no rotating stuff first

Hong Kong Olympiad in Informatics 2007 Senior Question 3 SOS

> Hackson Leung

Statistics

Problem

Description

Naïve Solutio

E00/ Calustan

70% Solution

100% Solution

- 1 The bottleneck is "How to find the nearest space to an element" efficiently
- Consider no rotating stuff first
 - Let *P* be a list storing the position of spaces in a row

Hong Kong Olympiad in Informatics 2007 Senior Question 3 SOS

> Hackson Leung

Statistics

Problem Description

Naive Solutio

50% Solution

70% Solution

1000/ 6-1-2

- The bottleneck is "How to find the nearest space to an element" efficiently
- Consider no rotating stuff first
 - Let *P* be a list storing the position of spaces in a row
 - e.g. bbsbsbbsbs

Hong Kong Olympiad in Informatics 2007 Senior Question 3 SOS

> Hacksor Leung

Statistics

Problem Description

. taire solatio

50% Solution

70% Solution

- 1 The bottleneck is "How to find the nearest space to an element" efficiently
- Consider no rotating stuff first
 - Let P be a list storing the position of spaces in a row
 - e.g. bbsbsbbsbs
 - $P = \{3, 5, 8, 10\}$

Hong Kong Olympiad in Informatics 2007 Senior Question 3 SOS

Hackson Leung

Statistic

Problem Description

Naive Solution

70% Solution

7070 30101101

- The bottleneck is "How to find the nearest space to an element" efficiently
- Consider no rotating stuff first
 - Let *P* be a list storing the position of spaces in a row
 - e.g. bbsbsbbsbs
 - $P = \{3, 5, 8, 10\}$
 - I want to know where is the nearest space with respect to the 6th element

Hong Kong Olympiad in Informatics 2007 Senior Question 3 SOS

Hackson Leung

Statistic

Problem Description

Naive Solution

70% Solution

70/0 301111011

- 1 The bottleneck is "How to find the nearest space to an element" efficiently
- Consider no rotating stuff first
 - Let *P* be a list storing the position of spaces in a row
 - e.g. bbsbsbbsbs
 - $P = \{3, 5, 8, 10\}$
 - I want to know where is the nearest space with respect to the 6th element
 - **3?** 5? 8? 10?

Hong Kong Olympiad in Informatics 2007 Senior Question 3 SOS

Hackson Leung

Statistic

Problem Description

Naive Solution

70% Solution

- The bottleneck is "How to find the nearest space to an element" efficiently
- Consider no rotating stuff first
 - Let *P* be a list storing the position of spaces in a row
 - e.g. bbsbsbbsbs
 - $P = \{3, 5, 8, 10\}$
 - I want to know where is the nearest space with respect to the 6th element
 - **3**? 5? 8? 10?
 - Obviously P is sorted

Hong Kong Olympiad in Informatics 2007 Senior Question 3 SOS

> Hacksor Leung

Statistics

Problem Description

Naïve Solution

70% Solution

Special Note

The bottleneck is "How to find the nearest space to an element" efficiently

Consider no rotating stuff first

■ Let *P* be a list storing the position of spaces in a row

e.g. bbsbsbbsbs

 $P = \{3, 5, 8, 10\}$

■ I want to know where is the nearest space with respect to the 6th element

3? 5? 8? 10?

Obviously P is sorted

3 Binary Search can be applied

Hong Kong Olympiad in Informatics 2007 Senior Question 3 SOS

> Hackson Leung

Statistic

Problem Description

Naïve Solution

70% Solution

70/6 301011011

- The bottleneck is "How to find the nearest space to an element" efficiently
- Consider no rotating stuff first
 - Let *P* be a list storing the position of spaces in a row
 - e.g. bbsbsbbsbs
 - $P = \{3, 5, 8, 10\}$
 - I want to know where is the nearest space with respect to the 6th element
 - **3**? 5? 8? 10?
 - Obviously P is sorted
- 3 Binary Search can be applied
- 4 How to deal with rotating stuff?

70% Solution - Implementation

Hong Kong Olympiad in Informatics 2007 Senior Question 3

Hackson

Statistics

Problem Description

Naïve Solutio

0% Solutio

70% Solution

100% Solutic

70% Solution - Implementation

```
Hong Kong
Olympiad in
Informatics
2007
Senior
Question 3
SOS
```

Hackson Leung

Statistics

Problem Description

Naïve Solution

50% Solution

70% Solution

Special Note

70% Solution - Implementation

```
Algorithm 70% Solution  \begin{array}{ll} \text{MinRotation}[][] \leftarrow \infty \\ \text{MinRotation}[x][y] \leftarrow 0 & //\exists x,y \rightarrow (x,y) \text{ is a space} \\ \text{for each character } (r,c) \text{ do} \\ \text{Update MinRotation}[r][c] \leftarrow \textit{lower\_bound}(P,c) \\ \end{array}
```

$$Answer \leftarrow \min_{1 \leq i \leq M} \{ \sum_{j=1}^{N} MinRotation[j][i] \}$$

70% Solution - Implementation

Hong Kong Olympiad in Informatics 2007 Senior Question 3 SOS

> Hacksor Leung

Statistics

Problem Description

ivalve Solution

70% Solution

100% Solutio

Special Note

70% Solution - Implementation

Algorithm 70% Solution
$$\begin{array}{ll} \text{MinRotation}[]] \leftarrow \infty \\ \text{MinRotation}[x][y] \leftarrow 0 & //\exists x,y \rightarrow (x,y) \text{ is a space} \\ \text{for each character } (r,c) \text{ do} \\ \text{Update MinRotation}[r][c] \leftarrow \textit{lower_bound}(P,c) \\ \end{array}$$

$$Answer \leftarrow \min_{1 \leq i \leq M} \{ \sum_{j=1}^{N} MinRotation[j][i] \}$$

lower_bound is a binary search function that returns the first position where value could be inserted without violating the ordering. (Ref.: http://www.sgi.com/tech/stl/lower_bound.html)

Hong Kong Olympiad in Informatics 2007 Senior Question 3

> Hacksor Leung

Statistics

Problem

Naïve Solutio

0% Solutio

70% Solution

100% Solutio

Hong Kong Olympiad in Informatics 2007 Senior Question 3

> Hackson Leung

Statistic

Problem
Description

Naïve Soluti

00/ 0-1-1

70% Solution

1070 Solution

Special Not

A bit easier to implement

Hong Kong Olympiad in Informatics 2007 Senior Question 3

> Hacksor Leung

Statistic

Problem
Description

Naïve Solution

E00/ Calustan

70% Solution

100% Solution

- A bit easier to implement
- Much efficient

Hong Kong Olympiad in Informatics 2007 Senior Question 3

> Hacksor Leung

Statistic

Problem Description

50% Solution

70% Solution

Caralal Nas

A bit easier to implement

Much efficient

Runtime Complexity: $\mathcal{O}(NMlgM)$

Note: $2^x = y \rightarrow x = lgy$

Hong Kong Olympiad in Informatics 2007 Senior Question 3

> Hackson Leung

Statistic

Problem Description

Naïve Solution

70% Solution

- A bit easier to implement
- Much efficient
- Runtime Complexity: $\mathcal{O}(NMlgM)$ Note: $2^x = y \rightarrow x = lgy$
- However, if *M* is sufficiently large, it will still risk TLE

Hong Kong Olympiad in Informatics 2007 Senior Question 3

> Hackson Leung

Statistics

Problem

Naïve Solutio

0% Solutio

70% Solution
100% Solution

Caralal Nati

Hong Kong Olympiad in Informatics 2007 Senior Question 3

> Hackson Leung

Statistic

Problem Descriptio

Naïve Solutio

50% Solution

70% Solution

100% Solution

Special Note

Still considering no rotation stuff

Hong Kong Olympiad in Informatics 2007 Senior Question 3

> Hacksor Leung

Statistic

Problem
Description

Naïve Solution

-00/ Calusta

70% Solutio

100% Solution

- Still considering no rotation stuff
 - If we know MinRotation[α], we can consider MinRotation[$\alpha+1$] immediately.

Hong Kong Olympiad in Informatics 2007 Senior Question 3 SOS

> Hacksor Leung

Statistics

Problem Description

Traine Solution

50% Solutio

70% Solution

100% Solution

- Still considering no rotation stuff
 - If we know MinRotation[α], we can consider MinRotation[$\alpha + 1$] immediately.
 - Suppose we know MinRotation[α] corresponds to the space located at $P[\beta]$ which is on its left, then MinRotation[$\alpha+1$] can be determined by one of the two "neighbouring" spaces.

Hong Kong Olympiad in Informatics 2007 Senior Question 3 SOS

> Hacksor Leung

Statistic

Problem Description

Naïve Solution

70% Salutia

100% Solution

- Still considering no rotation stuff
 - If we know MinRotation[α], we can consider MinRotation[$\alpha + 1$] immediately.
 - Suppose we know MinRotation[α] corresponds to the space located at $P[\beta]$ which is on its left, then MinRotation[$\alpha+1$] can be determined by one of the two "neighbouring" spaces.
 - $\alpha + 1 > P[\beta + 1] \Rightarrow MinRotation[\alpha + 1] = min(|\alpha + 1 P[\beta + 1]|, |\alpha + 1 P[\beta + 2]|)$
 - $\qquad \text{Otherwise,} \\ \text{MinRotation}[\alpha+1] = \min(|\alpha+1-P[\beta]|, |\alpha+1-P[\beta+1]|)$

Hong Kong Olympiad in Informatics 2007 Senior Question 3 SOS

Leung

Statistic

Problem Description

Naive Solution

70% Solution

100% Solution

- Still considering no rotation stuff
 - If we know MinRotation[α], we can consider MinRotation[$\alpha+1$] immediately.
 - Suppose we know MinRotation[α] corresponds to the space located at $P[\beta]$ which is on its left, then MinRotation[$\alpha+1$] can be determined by one of the two "neighbouring" spaces.
 - $\alpha + 1 > P[\beta + 1] \Rightarrow MinRotation[\alpha + 1] = min(|\alpha + 1 P[\beta + 1]|, |\alpha + 1 P[\beta + 2]|)$
 - Otherwise, $\text{MinRotation}[\alpha+1] = \min(|\alpha+1-P[\beta]|, |\alpha+1-P[\beta+1]|)$
- **2** How about when $P[\beta] > \alpha$?

Hong Kong Olympiad in Informatics 2007 Senior Question 3 SOS

Leung

Statistics

Problem Description

Naive Solution

70% Solution

100% Solution

- 1 Still considering no rotation stuff
 - If we know MinRotation[α], we can consider MinRotation[$\alpha+1$] immediately.
 - Suppose we know MinRotation[α] corresponds to the space located at $P[\beta]$ which is on its left, then MinRotation[$\alpha+1$] can be determined by one of the two "neighbouring" spaces.
 - $\alpha + 1 > P[\beta + 1] \Rightarrow MinRotation[\alpha + 1] = min(|\alpha + 1 P[\beta + 1]|, |\alpha + 1 P[\beta + 2]|)$
 - Otherwise, $\text{MinRotation}[\alpha+1] = \min(|\alpha+1-P[\beta]|, |\alpha+1-P[\beta+1]|)$
- 2 How about when $P[\beta] > \alpha$?
- **3** Rotation stuff?!?!

Hong Kong Olympiad in Informatics 2007 Senior Question 3 SOS

> Hacksor Leung

Statistic

Problem Description

Naive Solution

70% Solution

100% Solution

- Still considering no rotation stuff
 - If we know MinRotation[α], we can consider MinRotation[$\alpha+1$] immediately.
 - Suppose we know MinRotation[α] corresponds to the space located at $P[\beta]$ which is on its left, then MinRotation[$\alpha+1$] can be determined by one of the two "neighbouring" spaces.
 - $\alpha + 1 > P[\beta + 1] \Rightarrow MinRotation[\alpha + 1] = min(|\alpha + 1 P[\beta + 1]|, |\alpha + 1 P[\beta + 2]|)$
 - Otherwise, $\text{MinRotation}[\alpha+1] = \min(|\alpha+1-P[\beta]|, |\alpha+1-P[\beta+1]|)$
- 2 How about when $P[\beta] > \alpha$?
- 3 Rotation stuff?!?!
- Much much faster

Hong Kong Olympiad in Informatics 2007 Senior Question 3 SOS

> Hacksor Leung

Statistics

Problem Description

Naive Solution

70% Solution

100% Solution

- Still considering no rotation stuff
 - If we know MinRotation[α], we can consider MinRotation[$\alpha + 1$] immediately.
 - Suppose we know MinRotation[α] corresponds to the space located at $P[\beta]$ which is on its left, then MinRotation[$\alpha+1$] can be determined by one of the two "neighbouring" spaces.
 - $\alpha + 1 > P[\beta + 1] \Rightarrow MinRotation[\alpha + 1] = min(|\alpha + 1 P[\beta + 1]|, |\alpha + 1 P[\beta + 2]|)$
 - Otherwise, $\text{MinRotation}[\alpha+1] = \min(|\alpha+1-P[\beta]|, |\alpha+1-P[\beta+1]|)$
- 2 How about when $P[\beta] > \alpha$?
- Rotation stuff?!?!
- 4 Much much faster
- 5 Runtime Coimplexity?

Special Note

Hong Kong Olympiad in Informatics 2007 Senior Question 3

> Hacksor Leung

Statistics

Problem Description

Naïve Solutio

50% Solutio

70% Solution

Special Note

Hong Kong Olympiad in Informatics 2007 Senior Question 3 SOS

> Hacksor Leung

Statistics

Problem
Description

ivalve Solution

50% Solution

70% Solutio

Special Note

During Competition:

- Don't use cin and cout!
- 2 Avoid using standard string and single character input.

Special Note

Hong Kong Olympiad in Informatics 2007 Senior Question 3 SOS

> Hackson Leung

Statistics

Problem Description

Naïve Solution

70% Solution

Special Note

During Competition:

- Don't use cin and cout!
- 2 Avoid using standard string and single character input.

In HKOI Online Judge:

- 1 Time Limit is changed to 1 second
- 2 Input may critically affecting the runtime if you use an improper way.
- 3 Never use too much memory (Why?)
- 4 HKOI Online Judge only accept the best solution. Don't try to submit every solution you've written.