TWGSS
OI Team

2022/23
Team Formation Test

Editorial

Statistics

	Problem	Attempts	Full	Max	Mean	Std Dev	LQ	Median	UQ	Subtasks	First Solve	
TF231	School Corridor	16	30	30	19.563	10.241	7	21	30	7 | 15 / 14 | 11 / 9 | 6	Tsang Tsz Hong	00:09:45
TF232	Ridiculous Price	16	30	30	22.313	10.312	18	25	30	11 | 14 / 14 | 12 / 5 | 7	Tsang Tsz Hong	00:22:25
TF233	Seating Plan	15	40	40	24.653	14.753	10.2	17.2	40	2 | 15 / 5 | 10, 2 | 5 / 8 | 9, 3.2 | 4 / 15 | 7, 6 | 2 / 10 | 7, 4 | 2	Tsang Tsz Hong	00:46:29
TF234	Chemcraft Time	10	50	50	25.4	19.971	7	25.5	44	7 | 9 / 13 | 5 / 24 | 5 / 6 | 1	Tsang Tsz Hong	01:21:47
TF235	Problem Descriptor	10	70	70	20.5	25.041	8	8	17	8 | 9 / 9 | 3 / 8 | 2 / 17 | 2 / 28 | 2	Hui Ka Wai	03:02:18
TF236	King's Army	5	90	77	36.4	28.626	6.5	32	68.5	13 | 4 / 7 | 3 / 12 | 3 / 8 | 2 / 20 | 2 / 17 | 1 / 12 | 0 / 1 | 0		
TF237	Blackboard Escape	6	90	40	26.667	14.907	20	30	40	2 | 5 / 3 | 5 / 4 | 5 / 5 | 5 / 6 | 5 / 20 | 3 / 50 | 0		
		16	**400**	**240**	**115.05**	**72.412**	**66.2**	**103.5**	**184**			

School Corridor

TF23 I

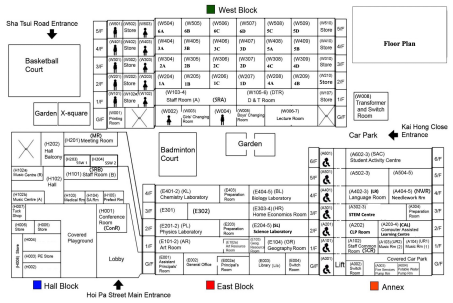
Story

Every Thursday, Daniel has to write Chinese essay during Chinese lesson, as Chinese writing is 1 hours 30 mins, but 2 lessons are only 1 hours 20 mins, so he has only 5 minutes left for the recess.

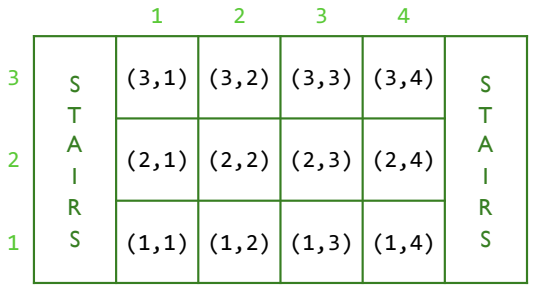
As Daniel needs going to the toilet, meeting friends, doing MC, etc., he is going to ***RUSH*** to the Biology Laboratory for the Biology lesson.

Problem Statement

We've simplified the floor plan of the school to simplified the problem.

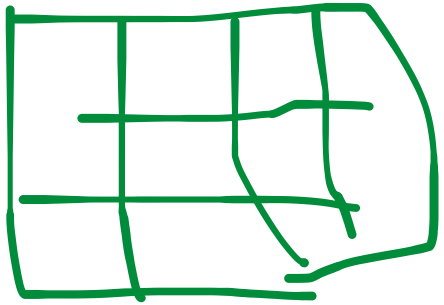


2019



2022

2030



2050

Problem Statement

Walking to the adjacent classroom/stair takes 1 second.

Walking up/down the stair takes 1 second.

ALL operations are 1s.

So you are going to find what is the minimum 1s you have to add to the answer.

Subtask 1

$H = 1$

The school only has ONE floor!

How easy is this!

Just output $\text{abs}(w_c - w_b)$!

DONE!

Score: 7

Subtask 2

$W = 2$

The school is only TWO classroom long! (but may have many floors tho)

How easy is this!

If $h_c = h_b$, the answer is $\text{abs}(w_c - w_b)$.

Else if $w_c = w_b$, the answer is $2 + \text{abs}(h_c - h_b)$.

Else the answer is $3 + \text{abs}(h_c - h_b)$.

Score: 14 (Cumulative: 21)

Full Solution

If $h_c = h_b$, the answer is $\text{abs}(w_c - w_b)$.

Else, consider taking left and right stair respectively.

For the left stair, time taken is $\text{abs}(h_c - h_b) + w_c + w_b$.

For the right stair, time taken is $\text{abs}(h_c - h_b) + (W - w_c + 1) + (W - w_b + 1)$.

Compare the two values and output the smaller one.

DONE!

Score: 30

Ridiculous Price

TF232

Story

Daniel wants to buy *lomien*, but the pricing is **ridiculous**!

This is the opposite of ridiculous price.

Problem Statement

Given N prices,

if any of the $\frac{a_j}{j} > \frac{a_i}{i}$ for $i < j$, output `Ridiculous`,

otherwise output `Normal`.

Subtask 1

$N = 2$
$1 \leq a_i \leq 100$

There are only 1 bag/2 bags of *lomien* prices!

Just test if $\frac{a_2}{2} > \frac{a_1}{1}$, output `Ridiculous` if true.

Otherwise output `Normal`.

Subtask 1

Why I get “Wrong Answer”?

Remember the C++ integer division?

It takes the rounded-down value!

Oops! So you have to use `double`!

Score: 11

Subtask 2

$1 \leq N \leq 10$
$1 \leq a_i \leq 100$

Use a for-loop of i looping through 2 to N:

If $\frac{a_i}{i} > \frac{a_{i-1}}{i-1}$ for any i, output `Ridiculous` and terminates the program.

Otherwise output `Normal`.

Score: 14 (Cumulative: 25)

Full Solution

There will be precision error using `double`!

So we have to use a more accurate approach.

Use a for-loop of i looping through 2 to N:

If $(i-1)\times a_i > i\times a_{i-1}$ for any i, output `Ridiculous` and terminates the program.

Otherwise output `Normal`.

Why? *The proof leaves as an exercise of the readers!*

Score: 30

Seating Plan

TF233

Problem Statement

Given a $N \times M$ grid and the number of student K you need to fit in.

Determine if it is possible to fit them in, if yes then output any possible layout.

Score 40% if output first line correctly.

Subtask 1

$N = 1$
$M = 2$
$K = 2$

Trivial that it's impossible to fit 2 student in 2 seats without them to sit adjacent to each other.

Therefore, just output `Impossible`.

Score: 2

Subtask 2

$K = 0$

How to fill 0 students in a $N \times M$ grid?

How?

How…?

How…….?????

You guess.

Subtask 2

It's always possible! *(Surprise surprise!)*

But remember to print the layout also, otherwise you will just have 40% score : (

Use nested for loop to print the layout.

Score: 5

Subtask 3

$N = M = 2$
$0 \leq K \leq 3$

We can consider each case separately (as there is only 4 of them).

When $K = 0$, you already solve it in Subtask 2.

When $K = 1$, you could put the only student anywhere you want!

When $K = 2$, you have 2 possible layout, just output anyone of them:

```
#.
.#
```

or

```
.#
#.
```

When $K = 3$, it's impossible!

Score: 8

Subtask 4

M is even

The very limited constraints are no longer here, so we cannot hardcode anymore!

Then how should we approach the subtask?

Slow down… and make some observation first.

Observation

We could observe the following patterns of maximum number of students that can be filled in:

Grid Size	4×1	4×2	4×3	4×4
Max. K	2	4	6	8

See the pattern?

Max. number of students that can be filled in is $N\times M \div 2$.

By pigeonhole principle, we can prove that it can't be greater than $N\times M \div 2$.

Subtask 4

Therefore, to determine if it is possible, we just need to check if $K \leq N \times M \div 2$.

40% score in bag!

How should we print the layout though?

Although you can make some crazy layout, it is better to make it as simple as possible.

For odd rows, we can put students in columns $1, 3, 5, 7, \cdots$.

For even rows, we can put students in columns $2, 4, 6, 8, \cdots$.

Score: 25

Full solution

Now M can be odd too!

N can be odd or even.

When N is even, the solution is just similar to that when M is even.

So we only need to consider when N and M are both odd, i.e. $N \times M$ is odd.

Full solution

We can consider the following examples:

Grid Size	3×3	5×5
Max. K	5	13

Layout	`#.#`
	`.#.`
	`#.#`
		`#.#.#`
		`.#.#.`
		`#.#.#`
		`.#.#.`
		`#.#.#`

Observed pattern: For odd $N \times M$, maximum K is $(N \times M + 1) \div 2$.

Full solution

Observed pattern: For odd $N \times M$, maximum K is $(N \times M + 1) \div 2$.

Therefore, we could construct the following logic:

- When $N \times M$ is odd, it is possible when $K \leq (N \times M + 1) \div 2$.
- Else , it is possible when $K \leq (N \times M) \div 2$.

We got 40% for subtask 5!

Now, we just have to construct the layout of the grid when $N \times M$ is odd.

Full solution

To construct the layout, you could just follow the same rule as Subtask 4.

For odd rows, we can put students in columns $1, 3, 5, 7, \cdots$.

For even rows, we can put students in columns $2, 4, 6, 8, \cdots$.

Problem solved!

Score: 40

Chemcraft Time

TF234

Problem Statement

The time used for crafting the i-th particle is $T_i + \sum_{j=1}^{K_i} p_{i,j}$

What does it mean?

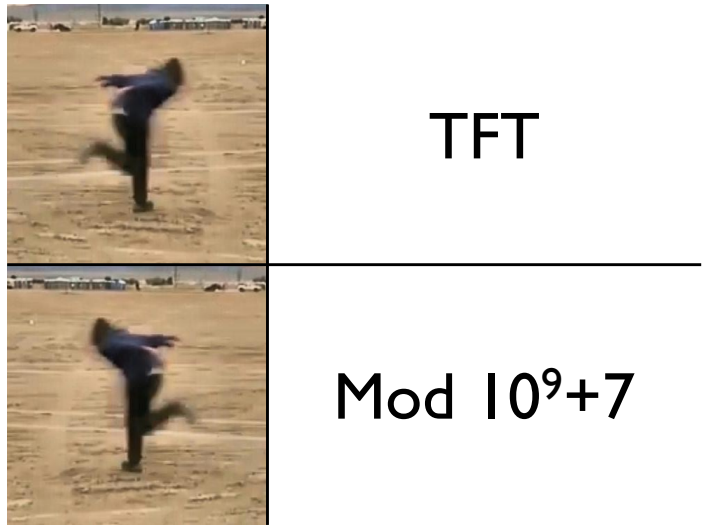
- Crafting ALL particles require the formation time T_i
- The crafting time is then added by the crafting time of the K_i (K_i maybe 0) particles. The K_i particles are represented as $p_{i,1}, p_{i,2}, \ldots, p_{i,K_i}$

Mod 10^9+7

Although mod 10^9+7 is mentioned, it is only applicable in *subtask 4*!

Note the input constraints!

So if you don't know how to do mod $10^9 + 7$, you can still happily attempt the problem.



Subtask 1

$K_i = 0$ for $1 \leq i \leq N$

Just output each T_i

Expected Score: 7

Subtask 2

$K_1 = 0, K_i = 1$ for $2 \leq i \leq N$

$p_{i,1} = i - 1$ for $2 \leq i \leq N$

$1 \leq T_i \leq 1000$

Input formed a “chain”

Partial sum!

Expected Score: 20

Subtask 3

$1 \leq N \leq 10$
$1 \leq T_i \leq 5$

All the items used to form itself is **simpler**

Therefore, just compute the formation time in order of increasing IDs (i.e. from simplest to most complicated)!

Time Complexity: O(N)

Expected Score: 44

Full Solution

Use the solution in Subtask 3

Remember to mod 10^9+7

Time Complexity: O(N)

Expected Score: 50

Problem Descriptor

TF235

Story

"What problem do you have?"

"What problem?"

"No, I ask you, what problem do you have?"

"No, what problem do you have, what problem do I have?"

"I ask you, what problem do you have?"

"Har?"

conversation

Daniel hears this legendary ~~conversion~~ after the Chemistry UT on bus route 42!

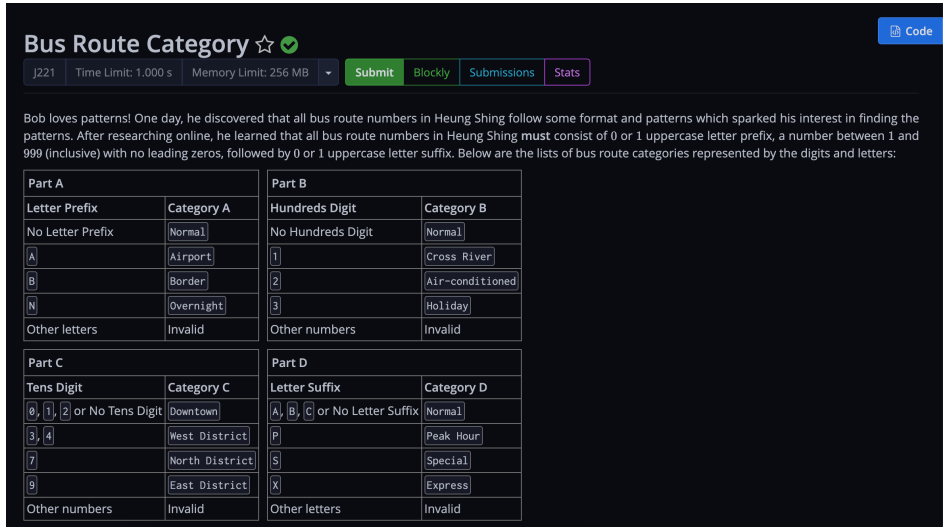
Story

Talking about *bus route*…

We'll think of the legendary HKOI problem – *Bus Route Category (J221)*!

So this problem is EXTREMELY similar to that task!

(Duncan said it was annoying)



Bus Route Category

Code

J221 | Time Limit: 1.000 s | Memory Limit: 256 MB | Submit | Blockly | Submissions | Stats

Bob loves patterns! One day, he discovered that all bus route numbers in Heung Shing follow some format and patterns which sparked his interest in finding the patterns. After researching online, he learned that all bus route numbers in Heung Shing **must** consist of 0 or 1 uppercase letter prefix, a number between 1 and 999 (inclusive) with no leading zeros, followed by 0 or 1 uppercase letter suffix. Below are the lists of bus route categories represented by the digits and letters:

Part A

Letter Prefix	Category A
No Letter Prefix	`Normal`
`A`	`Airport`
`B`	`Border`
`N`	`Overnight`
Other letters	Invalid

Part B

Hundreds Digit	Category B
No Hundreds Digit	`Normal`
`1`	`Cross River`
`2`	`Air-conditioned`
`3`	`Holiday`
Other numbers	Invalid

Part C

Tens Digit	Category C
`0`, `1`, `2` or No Tens Digit	`Downtown`
`3`, `4`	`West District`
`7`	`North District`
`9`	`East District`
Other numbers	Invalid

Part D

Letter Suffix	Category D
`A`, `B`, `C` or No Letter Suffix	`Normal`
`P`	`Peak Hour`
`S`	`Special`
`X`	`Express`
Other letters	Invalid

Problem Statement

Read it yourself.

Subtask 1

The problem ID has the prefix of `X`, `B`, `A` or `W` and is guaranteed to be valid

How easy is this!

Assume the problem ID be `string S`.

Just use an if-statement to test `S[0]` output the *problem type representation*.

Then just output `Problem S.substr(1, S.length() - 1)`.

(or you can just `cout << S[1] << S[2] << S[3];`)

DONE!

Score: 8

Subtask 2

The problem ID has the prefix of TF and is guaranteed to be valid

How easy is this!

First check the year using `S[2]` and `S[3]`,

```
int year = (S[2]-'0')*10 + (S[3]-'0');
```

If `year >= 22`, then output `20`, else output `21`;

then, just output `Team Formation Test`.

then output `year`.

Then output `Problem S[4]`.

DONE!

Score: 9 (Cumulative: 17)

Subtask 3

The problem ID has the prefix of `TW` and is guaranteed to be valid

How easy is this!

Frist output the *full school year representation*.

Then, you can use 9 if-statement or use an array to store all ordinal numbers to output the *full round number representation*.

Then, just output `Operation Imaginary`.

Then output `Problem S[5]`.

DONE!

Score: 8 (Cumulative: 25)

Subtask 4

The problem ID is guaranteed to be valid

How easy is this!

Just do the TC case!

I trust you can do it!

DONE!

Score: 17 (Cumulative: 42)

Full Solution

How easy is this!

Subtask 4 solution

+

Careful checking of whether the input is valid or not

DONE!

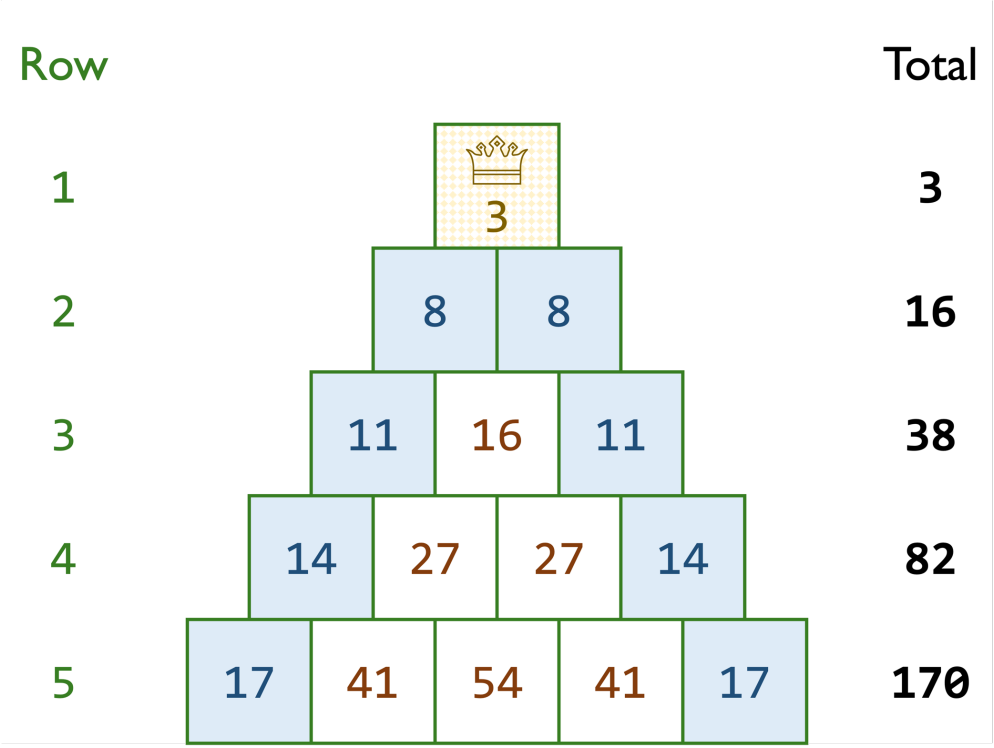
Score: 70

King’s Army

TF236

Problem Statement

Given the leadership level L and the base ability of the 2nd row soldiers B, find the total fighting ability of the required level for each queries.



Subtask 1-5

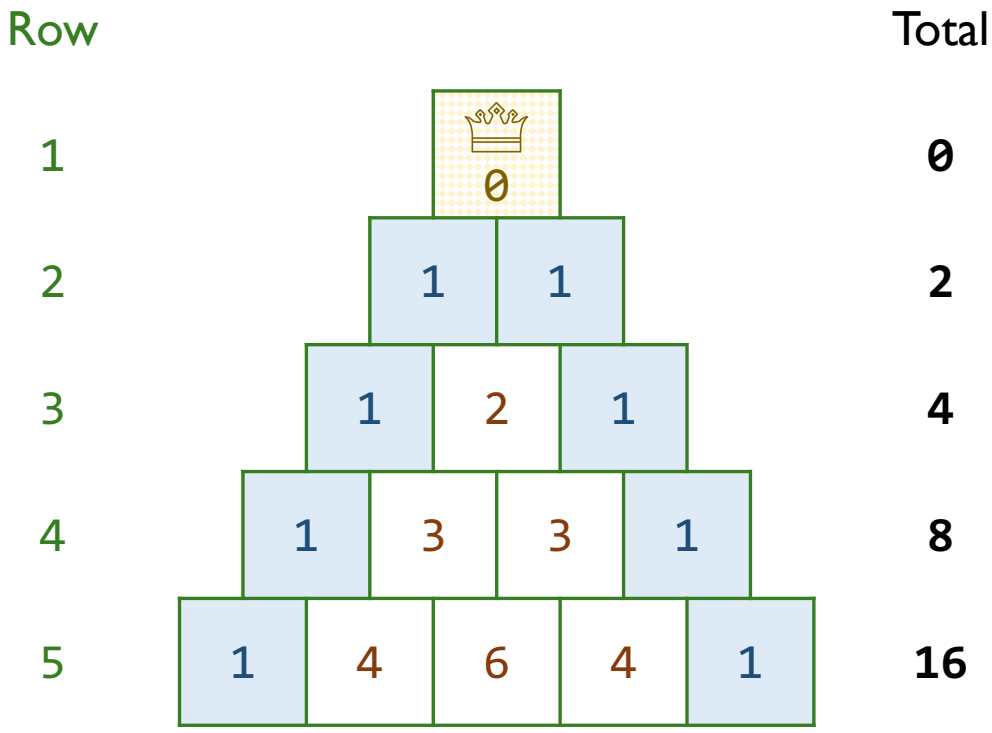
Just simulate it, here is a sample case:

```cpp
#include <bits/stdc++.h>
using namespace std;

int main() {
  long long a,b,c,d,g,bigmod = 1e9+7;
  cin >> a >> b >> c >> d;
  long long e[a][a],f[d];
  e[0][0] = b;
  e[1][0] = b+c;
  e[1][1] = b+c;
  for (int j=2;j<a;j++){
    for (int i = 0;i<=j;i++){
      if (i == 0){
        e[j][i] = e[j-1][i] + b;
      }else if(i == j){
        e[j][i] = e[j-1][i-1] + b;
      }else{
        e[j][i] = e[j-1][i-1] + e[j-1][i];
      }
    }
  }
  for (int i = 0;i<d;i++){
    cin >> f[i];
  }
  for (int i = 0;i<d;i++){
    g = 0;
    for (int j=0;j<f[i];j++){
      g += e[f[i]-1][j];
    }
    cout << g%bigmod << endl;
  }
}

```

Subtask 6

$1 \leq Q, R \leq 5000$

Observe the pattern when $L = 0, B = 1$:



See the pattern?

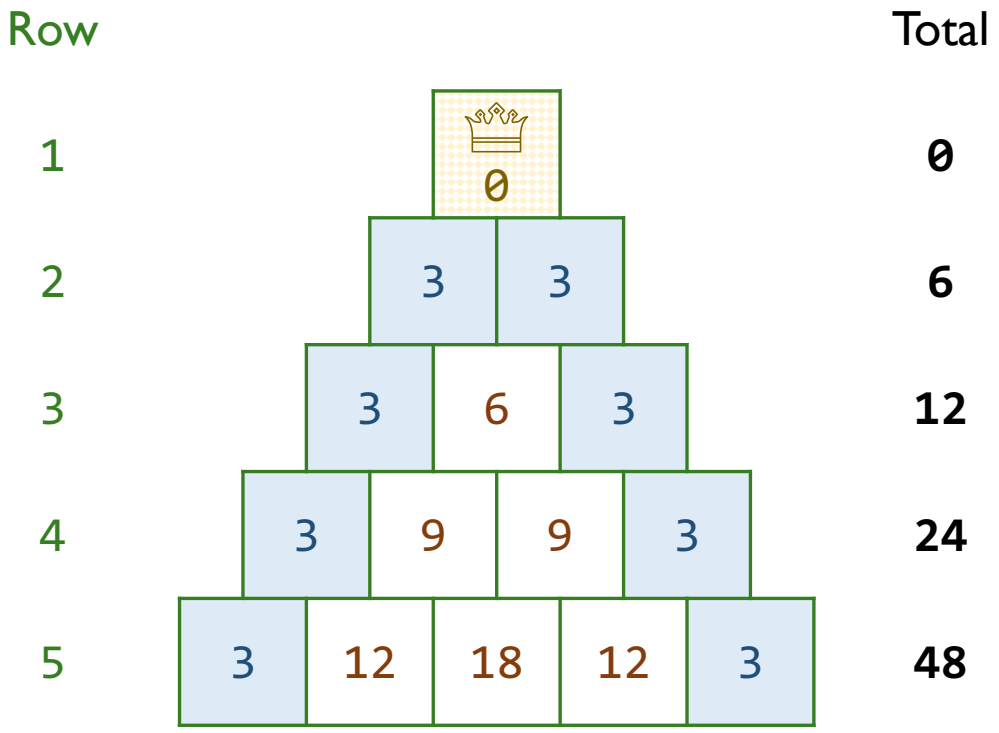
For row N $(N \neq 1)$,

the total fighting ability is 2^{N-1}.

Subtask 6

$1 \leq Q, R \leq 5000$

Observe the pattern when $L = 0, B = 3$:



See the pattern?

For row N $(N \neq 1)$,

the total fighting ability is $2^{N-1} \times 3$.

Thus we can prove that for $L = 0$,

the total fighting ability is $2^{N-1} \times B$.

Subtask 6

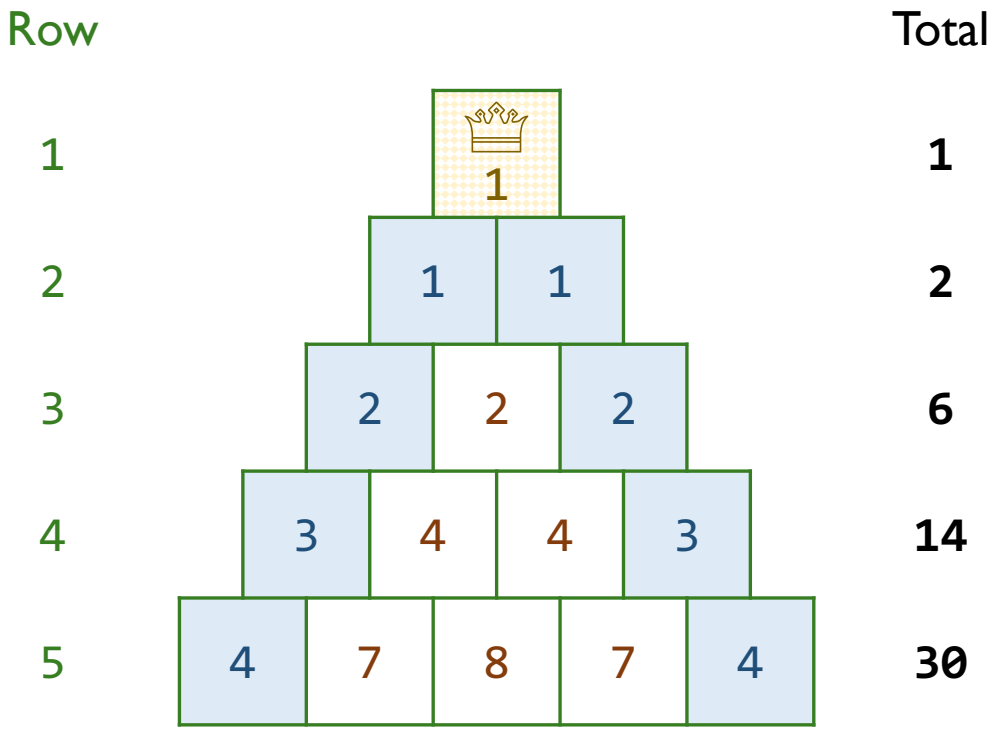
$1 \leq Q, R \leq 5000$

Now consider $L = 1, B = 0$:

See the pattern?

For row N $(N \neq 1)$,

the total fighting ability is $2^N - 2$.



Subtask 6

$1 \leq Q, R \leq 5000$

Now consider $L = 3, B = 0$:

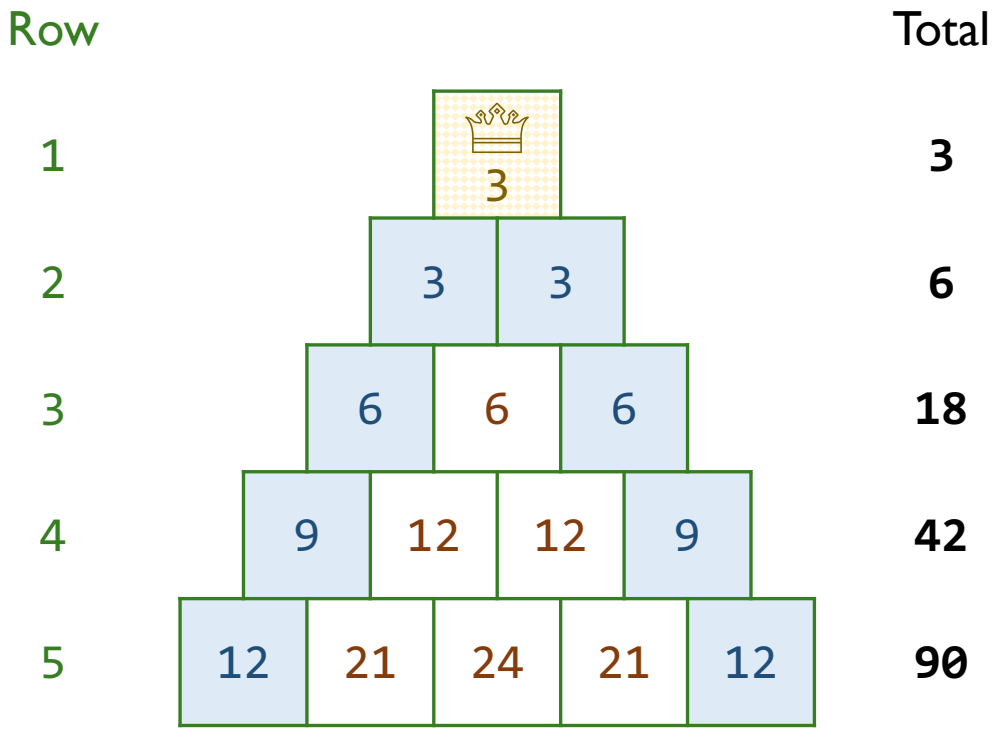
See the pattern?

For row N $(N \neq 1)$,

the total fighting ability is $(2^N - 2) \times 3$.

Thus we can prove that for $B = 0$,

the total fighting ability is $(2^N - 2) \times L$.



Subtask 6

$1 \leq Q, R \leq 5000$

When $L = 0$, total fighting ability $= 2^{N-1} \times B$.
When $B = 0$, total fighting ability $= (2^N - 2) \times L$.

Actually the total fighting ability affected L and B are independent from each other!

So the general formula is just adding them together!
We get $2^{N-1} \times B + (2^N - 2) \times L$ for row N $(N \neq 1)$.

Score: 77

Subtask 7

$1 \leq Q, R \leq 10^5$

We can just precompute the total fighting ability of each row. When the row is queried, we can just simply output the precomputed value.

Time complexity: O(R+Q)

Score: 89

Full Solution

If we use Subtask 7 solution, we will get TLE as the time complexity is O(R+Q), and R can be as large as 10^{18}.

Thus for each query, we can just calculate the answer using the general formula in Subtask 6: $(2^{N-1} \times B + (2^N - 2) \times L) \bmod 10^9 + 7$.

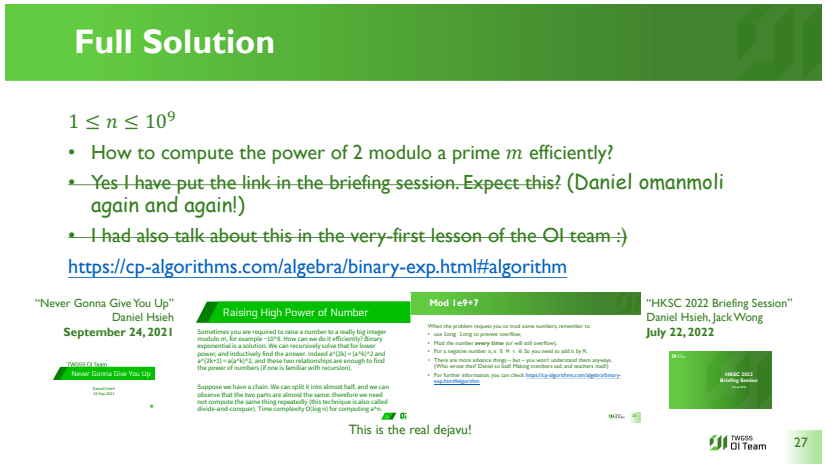
But how do we calculate the answer fast enough when N is large?

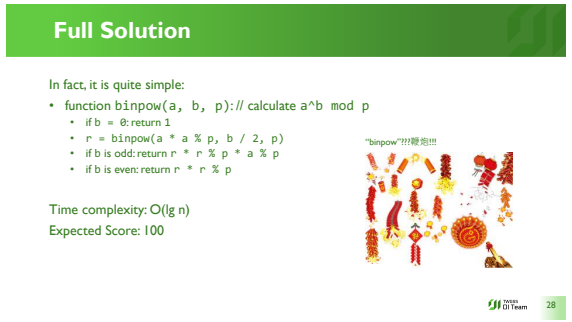
Binpow!

Time complexity: O(Q lgR)

Score: 90

Dejavu, yet again!





"2022 HKSC Practice Editorial"
July 24, 2022

Blackboard Escape

TF237

Problem Statement

Initially, only $0, 1$ are on the blackboard.

For each operation, you can choose two integers x, y on blackboard and write down $2y - x$.

You win when you write down N ($2 \leq N \leq 5 \times 10^8$).

Minimise the number of operations K used.

Subtask 1, 4

Subtask 1 $N = 2$

Subtask 4 $N = 5$

These cases are in the sample!

Do not be scared simply because this is Q7!

Expected Score: 2 + 5 = 7

Side Note: You can do the Subtasks 1-5 separately as the scoring is "cumulative"

Subtask 1-5

$2 \leq N \leq 6$

Do you understand the problem statement?

You should get these points quite easily!

Note that $x, y, 2y - x$ is just an arithmetic sequence: $(2y - x) - y = y - x$.

Therefore, there is intuitive reason to use $2y - x$ as the function.

You can do them by hand and obtain easy 20 points!

Expected Score: 20

Subtask 6

$2 \leq N \leq 10^5$

Induction!

Initially, there are only 0 and 1.

In operation 1, we may choose $x = 0, y = 1$, and write down $2y - x = 2$.

In operation 2, we may choose $x = 1, y = 2$, and write down $2y - x = 3$.

…

In operation $N - 1$, we may choose $x = N - 2, y = N - 1$, and write down N.

No. of operations: $N - 1$

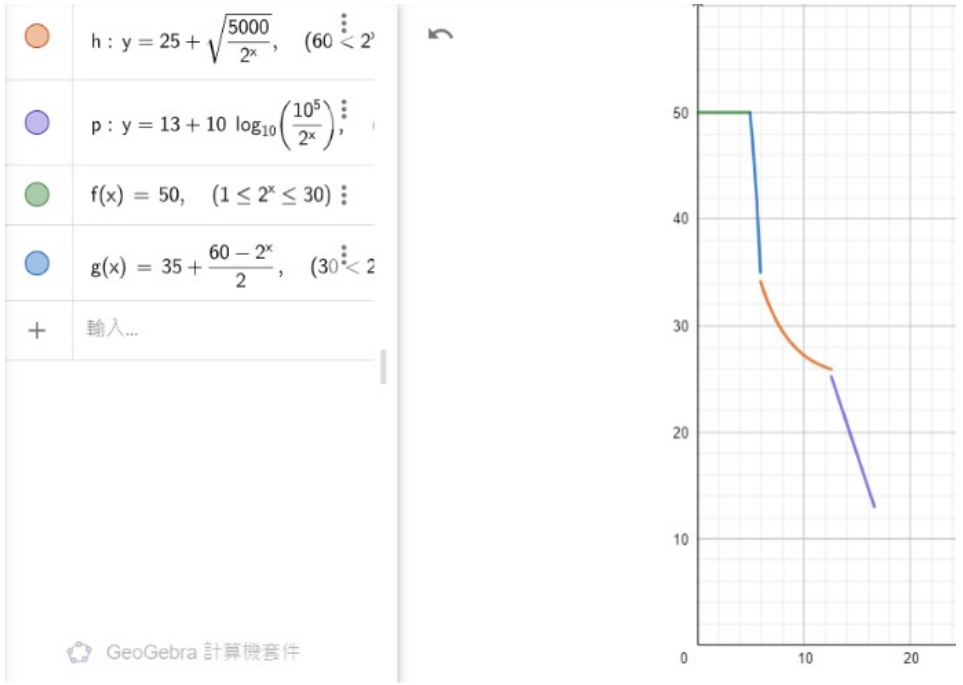
Expected Score: 40

Subtask 7

$2 \leq N \leq 5 \times 10^8$

Partial Scoring: the score obtained depends on the max no. of operations $K_{\max}$ over all test cases you used

$K_{\max} \leq 30$ for full score



Subtask 7a

$K_{\max} \leq 10^5$

How to do $2 \leq N \leq 5 \times 10^8$ using just 10^5 operations?

You might notice in our solution to Subtask 6, we always choose adjacent numbers.

But we can actually choose numbers farther away!

Let's say we first write down $1, 2, 3, \cdots, X$ using method in Subtask 6.

Next, we choose $N\%Y, N\%Y + Y$ $(N\%Y + Y \leq X)$ and again use method in Subtask 6 to write down $N\%Y + 2Y, N\%Y + 3Y, \cdots, N$.

If we choose $Y \sim \sqrt{N}$, then no. of operations $\sim \sqrt{N}$.

Expected Score (lower bound): 53

Subtask 7b

$K_{\max} \leq 6000$

Can we further optimize our solution?
We may use multiple steps to write down N.

Instead of two steps as in Subtask 7a, we may use three steps!
Let's say we first write down $1, 2, 3, \cdots, X$.
Next, we write down $\cdots, Y - 2X', Y - X', Y$.
Finally we write down $\cdots, N - 2Y', N - Y', N$.
X' and Y' depends on the size of X and Y.

No. of operations ~ $N^{\frac{1}{3}}$
Expected Score (lower bound): 65

Subtask 7c

$K_{\max} \leq 60$

How about we further generalise it?

Binary Number!

No. of operations $\sim 2 \lg N \sim 2 \times 29 \sim 60$

Expected Score: 75

```
10111
01111
00111
00101
00011
00010
00001
00000
```

Full Solution

$K_{\max} \leq 30$

Note that we may use negative integers!

Claim: If there are two integer x, y such that $N \bmod (y - x) = x$, we may construct a solution.

We may repeatedly change our x and y.
Every operation, we may change (x', y') to $(x, 2y - x)$ or $(2x - y, y)$.
The difference $y' - x'$ is doubled, where $N \bmod (2y - 2x)$ is either x or $2x - y$.

No. of operations: $\lg N \sim 29$
Expected Score: 90