TWGSS
OI Team

2023/24 Team Selection Test

Editorial

Felix Lo Jack Wong

3 July 2024

Statistics

	Problem	Attempts	Full	Max	Mean	Std Dev	LQ	Median	UQ	Subtasks	First Solve	
TS240	Relaxing Sunday	16	20	20	16.5	6.432	16	20	20	5 | 14 / 7 | 14 / 8 | 12	Wong Ho Ki	00:05:01
TS241	Character Repetition	12	30	30	21.5	12.835	9	30	30	14 | 9 / 4 | 9 / 8 | 8 / 4 | 8	Fung Hiu Nok	00:05:18
TS242	Fuk Luk Sau	14	40	40	20.857	18.466	0	20	40	6 | 10 / 12 | 7 / 6 | 7 / 10 | 7 / 6 | 6	Hui Ka Wai	00:31:08
TS243	Grand School Award	7	65	65	38.429	26.928	0	37	65	12 | 5 / 20 | 3 / 15 | 5 / 10 | 5 / 8 | 3	Fok Kin Wing	00:58:29
TS244	Tiers of the Kingdom	4	95	24	6	10.392	0	0	12	9 | 1 / 15 | 1 / 13 | 0 / 16 | 0 / 18 | 0 / 24 | 0		
		16	**250**	**179**	**61.5**	**59.699**	**7**	**47**	**127**			

Relaxing Sunday

TS240

Problem Statement

Given

the day of the week of the first day S in a year

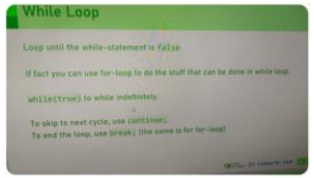
↓

Determine

whether the d^{th} day of the year is Sunday

Subtask 1

$d = 1$



呢條友係唔係唔識英文完全唔明佢嗡乜

Assume that you are not some OI Team trainers,
you should understand this if you understand English…
“If the first day of the year is Sunday, then the 1^{st} day of the year is Sunday.
Otherwise, the 1^{st} day of the year is not Sunday.”

Just use a single if-else statement to determine the output!
If S is `Sunday`, output `Relax`. Otherwise, output `Angry`.

Expected Score	5	Cumulative	5

Subtask 2

$S =$ Sunday

The first day of the year is Sunday.

When is the next Sunday?

The $8^{\text{th}}, 15^{\text{th}}, 22^{\text{nd}}, \cdots, 365^{\text{th}}$ day!

Can you find the pattern?

When $d = 7k + 1$ where k is integer, the queried day is Sunday.

In other words, only if $d - 1$ divisible by 7, the output is `Relax`.

In our code, we can check this by checking whether `(d-1) % 7` equals to `0`.

Expected Score	7	Cumulative	12

Full Solution

What about $S \neq$ `Sunday`?

Can we modify the condition a bit?

When $S =$ `Saturday`, $d = 7k + 2$ if it is Sunday.

When $S =$ `Friday`, $d = 7k + 3$ if it is Sunday.

$\vdots$ $\vdots$

Basically $d = 7k + x$ if the queried day is Sunday.

Thus only if $d - x$ divisible by 7, the output is `Relax`.

In our code, we can check whether `(d-x) % 7` equals to `0`.

S	x
`Sunday`	1
`Saturday`	2
`Friday`	3
`Thursday`	4
`Wednesday`	5
`Tuesday`	6
`Monday`	7

Expected Score	20	Cumulative	20

Common Errors

Compilation Error

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

int main(){
    string S,d;
    cin >> S;
    if(S == "Sunday"){
        cout << "Relax" << endl;
    }else if (d = 1){
        cout << "Relax" << endl;
        }else{
        cout >> "Angry" << endl;
    }

    }
}
```

d should be int

Too many }

Your code should pass ALL possible cases within the constraints!

Not just Sample Tests!

SAMPLE TESTS

	Input	Output
1	Monday 1	Angry

This sample satisfies the constraints of Subtask 1.

	Input	Output
2	Sunday 15	Relax

This sample satisfies the constraints of Subtask 2.

	Input	Output
3	Saturday 53	Angry
4	Wednesday 222	Relax

```
int main(){
  int d,n[100001];
    string S;
    cin>>S;
    cin>>d;
      if(S=="Sunday"&&(d-1)%7==0){
        S="Relax";
         cout<<S<<endl;
      }else if(S=="Wednesday"&&(d-1)%7==3){
        S="Relax";
         cout<<S<<endl;
      }else{
          S="Angry";
         cout<<S<<endl;
      }
  return 0;
}
```

Character Repetition

TS241

Problem Statement

Given

a string S which consists of English letters and spaces

↓

Delete consecutive repeated characters

such that every neighbouring characters are distinct

where the comparation is case-insensitive (e.g. A equals to a)

Subtask 1

S consists of lowercase English letters (`a`-`z`) only

We don't need to handle space or uppercase letters.

Therefore to solve this problem, we can…

- Input the string by using `cin`
- Use a for-loop with `i` ranging from `0` to `S.size()-1` to check repetition

  For each iteration, output `S[i]` if `S[i] != S[i-1]`

  Note that you should output `S[0]` directly when `i = 0` as there is no `S[-1]`

Expected Score	14	Cumulative	14

Subtask 2

S consists of lowercase English letters (a-z) and spaces only

With spaces in S, `cin` no longer works!

We can use `getline(cin, S)` to get S.

Then just apply Subtask 1's for-loop.

2023/24 C++ Foundation Course
Chapter 6 - Arrays

Expected Score	18	Cumulative	18

Subtask 3

S consists of English letters (`A-Z`, `a-z`) only

As the comparation is case-insensitive, we can first write a function that converts characters to lowercase before checking whether they the same.

```
char low(char c) {
    if (c >= 'A' && c <= 'Z')          check if c is uppercase
        c = c - 'A' + 'a';             change c to lowercase
    return c;
}
```

Subtask 3

S consists of English letters (`A-Z`, `a-z`) only

Adding case-insensitive comparison to Subtask 1 solution…

- Input the string by using `cin`
- Use a for-loop with `i` ranging from `0` to `S.size()-1` to check repetition
  - For each iteration, output `S[i]` if `low(S[i]) != low(S[i-1])`
  - Note that you should output `S[0]` directly when `i = 0` as there is no `S[-1]`

Expected Score	22	Cumulative	26

Full Solution

Combining the ideas of Subtask 2 and Subtask 3...

- Using `getline()` for input
- Implement case-insensitive comparison

That's it.

Expected Score	30	Cumulative	30

Common Errors

- You cannot delete a character in a string by setting it to `0`

  and you should test using Sample Tests!

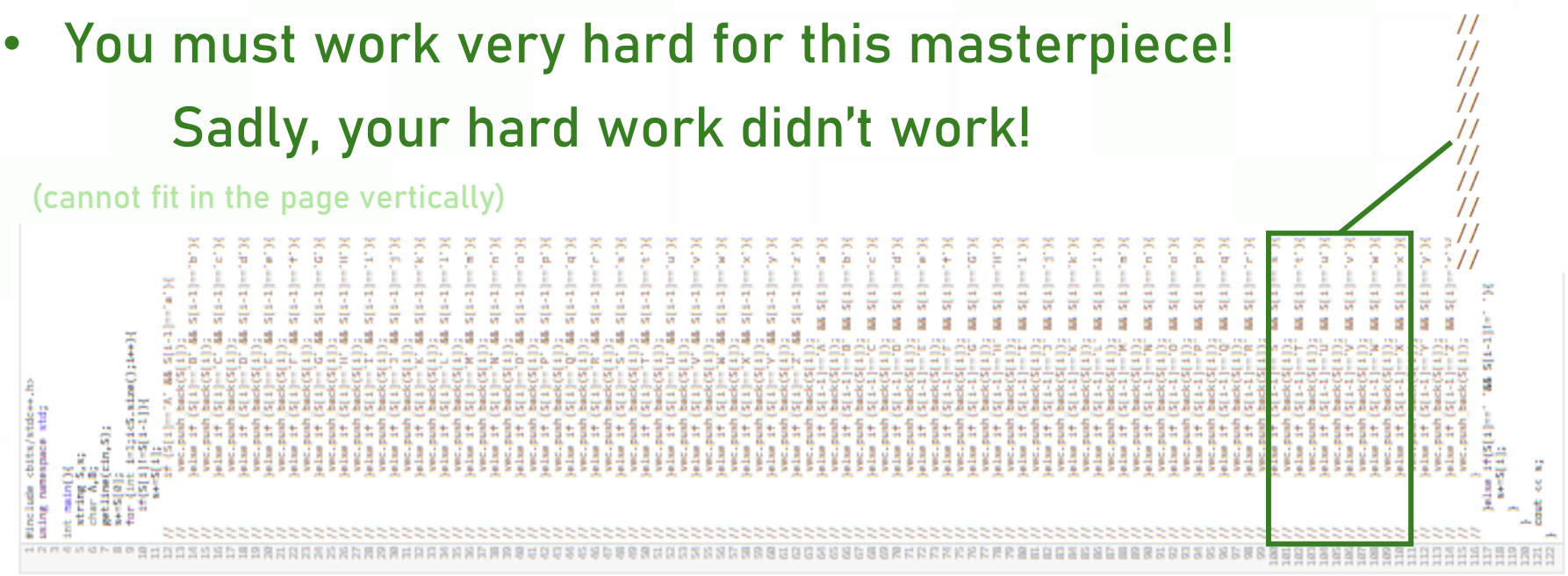
INPUT	OUTPUT
1 halllooooo	1 hal••o•••

```
int main(){
  string str;
    cin >> str;
    for(int i=str[str.length()-1]; i>0; i--){
        if(str[i]==str[i-1]){
            str[i]=0;
        }
    }
    cout << str;
}
```

- You must work very hard for this masterpiece!

  Sadly, your hard work didn't work!

(cannot fit in the page vertically)



```
//      }else if (S[i-1]=='S' && S[i]=='s'){
//       vec.push_back(S[i]);
//      }else if (S[i-1]=='T' && S[i]=='t'){
//       vec.push_back(S[i]);
//      }else if (S[i-1]=='U' && S[i]=='u'){
//       vec.push_back(S[i]);
//      }else if (S[i-1]=='V' && S[i]=='v'){
//       vec.push_back(S[i]);
//      }else if (S[i-1]=='W' && S[i]=='w'){
//       vec.push_back(S[i]);
```

Fuk Luk Sau

TS242

Problem Statement

N_F Fuk items

Package Price	P_1	P_2	$\cdots$	F_{N_F}
Original Price	F_1	F_2	$\cdots$	F_{N_F}

N_L Luk items

Original Price	L_1	L_2	$\cdots$	L_{N_L}

N_S Sau items

Original Price	S_1	S_2	$\cdots$	S_{N_S}

Answer is $\max(\left(F_x + L_y + S_z\right) - P_x,\ 0)$

Subtask 1

$N_F = N_L = N_S = 1$
$1 \leq P_i \ F_i \ L_i \ S_i \leq 5000$

From the problem statement, we know that

answer is $\max((F_x + L_y + S_z) - P_x, \ 0)$

In this subtask, $x = y = z = 1$

By substituting into the formula, answer is just $\max((F_1 + L_1 + S_1) - P_1, \ 0)$

Expected Score	6	Cumulative	6

Subtask 2

$1 \leq N_F \ N_L \ N_S \leq 20$
$1 \leq P_i \ F_i \ L_i \ S_i \leq 5000$

We can just brute force by trying all combinations!

- Create variable $ans = 0$
- Use three for-loops with $1 \leq x \leq N_F$, $1 \leq y \leq N_L$, $1 \leq z \leq N_S$

  For each iteration, update $ans = \max(\left(F_x + L_y + S_z\right) - P_x, \ ans)$

O(N³)	Expected Score	18	Cumulative	18

Subtask 3

$1 \leq N_F \ N_L \ N_S \leq 2000$
$1 \leq P_i \ F_i \ L_i \ S_i \leq 5000$

The package price actually depends on the Fuk item chosen!

So the selection of Luk item and Sau item can be performed separately!

Now we can rewrite the answer formula to $\max(F_x + \max(L_y + S_z) - P_x, \ 0)$

- Use two for-loops with $1 \leq y \leq N_L$, $1 \leq z \leq N_S$ to find $\max(L_y + S_z)$
- Use a for loop with $1 \leq x \leq N_F$ to find $\max(F_x + \max(L_y + S_z) - P_x, \ 0)$

O(N^2)	Expected Score	24	Cumulative	24

Subtask 4

$1 \leq P_i \ F_i \ L_i \ S_i \leq 5000$

We can consider the selection of Luk item and Sau item independently!

Rewrite the answer formula to $\max(\max(F_x - P_x) + \max(L_y) + \max(S_z),\ 0)$

- Just calculate each part separately by each using a for-loop
- Sum up $\max(F_x - P_x)$, $\max(L_y)$, $\max(S_z)$ and compare the result with 0

O(N)	Expected Score	34	Cumulative	34

Full Solution

As $1 \leq P_i \ F_i \ L_i \ S_i \leq 10^9$, answer can be as large as 3×10^9!

Therefore may I introduce our old friend – `long long`!

Just implement Subtask 4 solution with `long long`, done!

O(N)	Expected Score	40	Cumulative	40

Common Errors

- `long long` classics!

  → lose 6 marks (which is already very lenient!)

```
int F, L, S, Pac[200000], Fuk[200000], Luk[200000], Sau[200000], ans=0;
```

- Didn't output 0 when $(F_x + L_y + S_z) - P_x$ is negative

  → In order to remind you to output 0, you'll get 0 marks!

```
int main(){
  int a,b,c,d,f,l,s,t;
    cin>>a>>b>>c;
    cin>>d>>f>>l>>s>>t;
    t=f*a+l*b+s*c-d;
    cout<<t<<endl;
  return 0;
}
```

```
int main(){
    int a=1 ,b=1,c=1 ,d,e,f,g,h;
    cin >> a >> b >> c;
    cin >> d;
    cin >> e;
    cin >> f;
    cin >> g;
    cout << e + f + g -d << endl;
}
```

Grand School Award

TS243

Problem Statement

N data entries

For the i^{th} entry, the score P_i of the school S_i in the year Y_i is given

↓

Ranking the schools for every year

↓

Q queries

For the j^{th} query, output the rank of the school s_j in the year y_j

Subtask 1

$1 \leq N, P_i \leq 5000$
$Q = 1$

$S_x = s_1$ for exactly one x
$Y_1 = Y_2 = \cdots = Y_N = y_1$

Only one year with one guaranteed valid query!

- Use any data structure that can store pairs of integer (P) and string (S)
  e.g. `pair<int, string> A[]` or `vector<pair<int, string>> B`
- Sort the data structure (note that `pair` will sort by `.first`)
- Loop through the data structure to find the position of the queried entry

Subtask 1

$1 \leq N, P_i \leq 5000$
$Q = 1$

$S_x = s_1$ for exactly one x
$Y_1 = Y_2 = \cdots = Y_N = y_1$

Why my program outputs 2 when 5 should be outputted in Sample Test 1?

Remember that C++ `sort()` is default to sort in ascending order, which will make the rank to be ordered in reverse! Either way can solve this problem:

- `reverse()` the data structure after sorting
- write our own compare function which sorts in descending order and put it in the `sort()` function

O(N lgN)	Expected Score	12	Cumulative	12

Subtask 2

$$Y_1 = Y_2 = \cdots = Y_N = y_1 = y_2 = \cdots = y_Q$$

Only one year!

Similar to Subtask 1, we can just…

- Use any data structure that can store pairs of integer (P) and string (S)
  e.g. `pair<int, string> A[]` or `vector<pair<int, string>> A`
- Sort the data structure (note that `pair` will sort by `.first`)

Then we can just…

- Loop through the data structure to find the position of the queries

or can we…?

Subtask 2

$$Y_1 = Y_2 = \cdots = Y_N = y_1 = y_2 = \cdots = y_Q$$

We cannot “loop through the data structure” for every queries as this is a O(NQ) move!

Instead, we can pre-compute things!

- Use `map<string, int>` for storing S and their respective rank
- Loop through the data structure once for storing the answers in the `map`
- Just take the answers from the `map` for each query

Remember to output -1 when the query is not found in the `map`!

e.g. for a `map` named `C`, `!C.count(x)` or `C.find(x) == C.end()` will returns `true` if `x` is not found in it

O(N lgN + Q)	Expected Score	32	Cumulative	32

Subtask 3

$1 \leq N, Q, Y_i, P_i, y_j \leq 5000$

Limiting number of data and queries, and the year is bounded to 5000.

We can easily extend Subtask 1 solution for this subtask.

- Using an array of the original data structure
  e.g. `pair<int, string> A[][]` or `vector<pair<int, string>> B[]`
- Sort every data structure
  e.g. for $1 \leq k \leq 5000$, `sort(A[k], A[k]+N)` or `sort(B[k].begin(), B[k].end())`
- For each query, loop through the data structure of the respective year to find the position of the queried entry and output -1 when not found
  e.g. loop through $A[y_j]$ or $B[y_j]$ for the j^{th} query

O(N lgN Q)	Expected Score	15	Cumulative	47

Subtask 4

$1 \leq N, Q \leq 5000$

The year is not bounded to 5000, how can we tackle weirdly large years?

Oh! Here comes `map` again!

We can simply implement it by editing a bit of the Subtask 3 solution:

- Using ~~an array~~ a map of the original data structure
  e.g. ~~`map<int, pair<int, string>> A[]`~~ (not working anymore!) or
  `map<int, vector<pair<int, string>>> B`
- Sort every data structure
  Remember how to loop through every element in a `map`
- For each query, the handling remains unchanged

O(N lgN Q)	Expected Score	37	Cumulative	57

Full Solution

In Subtask 2, we implement pre-computing to prevent O(NQ).

In Subtask 3 and 4, we implement the solution for multiple years.

For the full solution, we can combine both progress into a definitive code!

- For the pre-computation, we can use `map<int, map<string, int>>` for storing different years' S and their respective rank

The remaining is just pure simple implementation!

Alternatively, you can use `struct` instead of the complicated data structures. That will keep your code cleaner and more readable!

```
struct sch{
  int yr, sc;
  string name;
}a[200005], tmp;
```

O(N lgN + Q)	Expected Score	65	Cumulative	65

Tiers of the Kingdom

TS244

Problem Statement

N tiers of floating islands

Tier i islands all have power level P_i

↓

You can merge K tier i islands to one tier $i+1$ island

Find maximum total power level by merging islands optimally

Observation 1

Merge K tier i islands to a tier $i+1$ islands changes the total power level by $P_{i+1} - KP_i$.

↓

If merging islands makes a positive change, we would want to merge as much islands as we can.

Subtask 1

$N = 2$

There are only two possible situations:

- Merging $\left\lfloor \frac{A_1}{K} \right\rfloor K$ tier 1 islands to $\left\lfloor \frac{A_1}{K} \right\rfloor$ tier 2 islands
- No merge at all

We just have to check which of the two possible situations gives a higher total power level and output its value.

Expected Score	9	Cumulative	9

Subtask 2

$N = 3$

Can we do this greedily by merging tier i islands to tier $i + 1$ islands only when it will return an instant profit ($P_{i+1} > KP_i$)?

No!

Note that we may encounter cases like this:

$K = 5$

i	1	2	3
P_i	**1**	**3**	**100**
A_i	**15**	**12**	**0**

Correct output `300`

Merging tier 1 islands to tier 2 islands seems stupid at first glance.

But if we merge tier 1 to tier 2 first, we have a total of 15 tier 2 islands, which can be further merged into 3 tier 3 islands.

Subtask 2

$N = 3$

Therefore, we have to consider all four situations:

- Merging tier 1 islands to tier 2 islands,
  then merging tier 2 islands to tier 3 islands
- Merging tier 1 islands to tier 2 islands only
- Merging tier 2 islands to tier 3 islands only
- No merge at all

Expected Score	15	Cumulative	24

Subtask 3

$A_1 = K$

$0 \leq A_i < K$ for $1 < i \leq N$

There is only one merge available initially.

After merging tier 1 islands to a tier 2 island,

then if $A_2 = K - 1$, we can choose to merge again to a tier 3 island,

then if $A_3 = K - 1$, we can choose to merge again to a tier 4 island...

Basically for any $1 < i < N$, if all $A_j = K - 1$ where $1 < j \leq i$,
we can choose to merge all islands from tier 1 to i to a tier $i + 1$ island.

Subtask 3

$A_1 = K$

$0 \leq A_i < K$ for $1 < i \leq N$

- Merge islands from low tier to high tier whenever possible
- Calculate the current total power level for each merge
- Keep track of the maximum total power level ever appeared

Optimally, we can make use of partial sum or other method to skip the calculation each merge, speeding up the code significantly.

O(N^2 or N)	Expected Score	13	Cumulative	37

Subtask 5

$0 \leq A_i \leq K$

We can split the array into several parts such that each array fulfil the constraints of Subtask 3.

We can solve each array using Subtask 3 method and add the answers together.

Remember that after splitting the array, you can still merge N islands at the end of each array to the first island of the next array.

O(N^2 or N)	Expected Score	31	Cumulative	55

Observation 2

There are two types of merging:

- Merging K^x tier i islands to a tier $i + x$ island
- Merging all tier $i + 1$ to tier $i + x - 1$ islands and some tier i islands to a tier $i + x$ islands

The first type of merge is worth if and only if the second type of merge is also worth.

Subtask 4

$1 \leq N \leq 10$

As N is small, we can loop try all possible ways to merge the islands and check whether they are worth.

Note that you would like to merge tier $i+1$ island first as tier i islands can be merged to tier $i+1$ island but tier $i+1$ islands cannot be split.

O(N^3)	Expected Score	40	Cumulative	71

Full Solution

There are N^2 ways to merge.

Therefore we have to calculate the cost in O(1) to prevent getting TLE.

- $C_{i\ldots i+n} = M_{i\ldots i+n} \times P_i + A_{i+1} \times P_{i+1} + \cdots + A_{i+n-1} \times P_{i+n-1}$
- $C_{i\ldots i+n+1} = M_{i\ldots i+n+1} \times P_i + A_{i+1} \times P_{i+1} + \cdots + A_{i+n} \times P_{i+n}$

$$= C_{i\ldots i+n} + A_{i+n} \times P_{i+n} + (M_{i\ldots i+n+1} - M_{i\ldots i+n}) \times P_i$$

where $M_{i\ldots i+n} + A_{i+1} \times K + \cdots + A_{i+n-1} \times K^{n-1} = K^n$

If you memorise $C_{i\ldots i+n}$, the cost can be calculate in O(1)!

O(N²)	Expected Score	95	Cumulative	95