

**Nama : Fajar Satria**

**NRP : 05111940000083**

**Kelas : B**

**Resume Kuliah PAA – 7 Juni 2021**

Pada pertemuan kali ini, ditegaskan kembali bahwa Greedy merupakan bagian dari divide and conquer dan juga greedy merupakan solusi dari persoalan-persoalan optimasi. Hal yang harus diperhatikan dalam greedy adalah pemahaman tentang local optimal yang bisa dibawa ke global optimal. Berikut adalah beberapa soal yang berkaitan dengan Greedy algorithm.

**a. E-olymp 10646 - Script**

**PERMASALAHAN**

Temirulan is an advanced programmer and often uses various scripts to do routine tasks. Sometimes instead of writing new scripts, his scripts are just sequences of scripts he wrote before.

Currently, Temirulan is working with NOOB (Network Optimized Object Base) online machine which accepts two types of requests:

1. Upload a script file and place it at the end of the machine's buffer for one dollar.
2. Copy the subsequent scripts in the machine's buffer and place them at the end of the buffer with no fee.

For simplicity, let's represent the sequence of scripts that Temirulan wants to execute with the string  $s$  of lowercase English letters. Each symbol represents a script file. Find the minimum amount of money Temirulan has to pay to run his script on the machine.

**SOLUSI**

Permasalahan inti pada soal ini adalah menghitung banyak macam karakter. Hal tersebut dikarenakan untuk tiap karakter yang sama, hanya karakter yang pertama yang dikenai biaya.

Perspektif solusi persoalan ini adalah menyimpan tiap karakter yang dimasukkan secara dinamis. Jika karakter yang dimasukkan disimpan secara statis maka akan kuwalahan jika testcase soal yang dipakai tidak hanya lowercase. Untuk melakukan penyimpanan secara dinamis tersebut dapat memakai Struktur Data Set.

Sehingga solusinya adalah mengiterasi tiap karakter yang diinputkan. Jika karakter tersebut belum ada pada Struktur Data Set, maka lakukan insert. Setelah selesai, tampilkan size dari Struktur Data tersebut. Hal ini dapat dilakukan karena size dari Struktur Data tersebut sama dengan banyak macam karakter yang sudah di insert kan.

**PSEUDOCODE**

SOLVE( $S$ )

1. let SC be a new Set
2. for  $i=0$  to  $S.length-1$
3.       //if character  $S[i]$  not present in SC Set then insert  $S[i]$  to SC Set
4.       if(!SC.count( $S[i]$ ))
5.               SC.insert( $S[i]$ )
6.   //return size of SC Set
7. return SC.size

### SOURCECODE

```
#include <set>
#include <string>
#include <cstring>
#include <iostream>
#include <algorithm>
using namespace std;
int main() {
    ios::sync_with_stdio(0);
    cin.tie(0); cout.tie(0);
    string s; cin >> s;
    set<char> sc;
    for(int i=0; i<s.length(); i++) {
        if(!sc.count(s[i])){
            sc.insert(s[i]);
        }
    }
    cout << sc.size();
    return 0;
}
```

### SUBMISSIONS

Solusi dinamis menggunakan Struktur Data Set memerlukan memory sebanyak **0.59MiB**. Sedangkan solusi statis tanpa menggunakan Struktur Data Set hanya memerlukan memory sebanyak **0.07MiB**. Namun solusi statis memiliki banyak kekurangan ketika testcase-nya adalah unicode.

Problem

Scout

Submitted

4 hours ago

Programming Language

C++ 17 (gnu 10.2)

Author

Fajar Setiyo\_0511194000093

100%

1 ms

0.59 MiB

Test #	Status	Score	Duration	CPU	Memory
✓ Test suite #1	Accepted	170 / 170	1 ms	1 ms	604 KiB
✓ Test #1	Accepted	10 / 10	1 ms	1 ms	596 KiB
✓ Test #2	Accepted	10 / 10	1 ms	1 ms	600 KiB
✓ Test #3	Accepted	10 / 10	1 ms	1 ms	596 KiB

Problem

Scout

Submitted

7 hours ago

Programming Language

C++ 17 (gnu 10.2)

Author

Fajar Setiyo\_0511194000093

100%

1 ms

0.07 MiB

Test #	Status	Score	Duration	CPU	Memory
✓ Test suite #1	Accepted	170 / 170	1 ms	1 ms	72 KiB
✓ Test #1	Accepted	10 / 10	1 ms	1 ms	72 KiB
✓ Test #2	Accepted	10 / 10	1 ms	1 ms	72 KiB
✓ Test #3	Accepted	10 / 10	1 ms	1 ms	72 KiB

### b. E-Olymp 66 – The Directors Visitors

#### PERMASALAHAN

The secretary of school Martha Georgiyivna starts her every working day with pretension to director:

- Here you, Ivan Ivanovych, already bought the program for scheduling to the deputy from educational part. And what do I have to do? I need to make the graphic of visitors according to Your needs in fact. And you didn't buy the program for work planning for me.

Please try to help secretary in her work. You have to organize the graphic of the visitors based on their wishes left in the secretary book.

Receiving two visitors at the same time is prohibited. When meeting with one visitor is finished, the meeting with another visitor can start – they can meet at the cabinet door.

#### SOLUSI

Permasalahan pada soal ini cukup mudah dipahami. Yakni menyusun jadwal sedemikian hingga agar mendapatkan jumlah meeting maksimum. Suatu jadwal tidak boleh saling overlapping antara satu dengan yang lain.

Sehingga solusi yang memungkinkan adalah menyimpan tiap jadwal kedalam sebuah Struct, dimana Struct tersebut berisi waktu mulai dan waktu berakhir. Dilanjutkan dengan melakukan sorting berdasarkan waktu berakhir. Untuk jadwal yang memiliki waktu berakhir yang sama, dilanjutkan dengan sorting berdasarkan waktu mulai.

Setelah selesai dengan sorting, hal yang harus dilakukan adalah mengiterasi tiap jadwal. Dimulai dari jadwal yang ke-2 dengan variable temp adalah waktu berakhir jadwal ke-1, dilakukan pengecekan “jika waktu mulai jadwal ke-2 lebih besar sama dengan waktu berakhir jadwal ke-1” maka geser variable temp menjadi waktu berakhir jadwal sekarang. Dilanjutkan sampai jadwal yang ke-n. Solusi akhirnya adalah banyaknya pergeseran variable temp.

### PSEUDOCODE

```
CMP(x,y)
1. If(x.finish == y.finish)
2.     return x.start < y.start
3. else    return x.finish < y.finish
SOLVE()
1. read n
2. let meets[0..n] be a new Struct
3. for i=0 to n-1
4.     read h1, m1, h2, m2
5.     meets[i].start = h1*60 + m1
6.     meets[i].finish = h2*60 + 21
7. sort(meets, meets+n, CMP)
8. ans = 1
9. temp = meets[0].finish
10. for i=1 to n-1
11.     if meets[i].start >= temp
12.         ans = ans + 1
13.         temp = meets[i].finish
14. print ans
```

### SOURCECODE

```
#include <cstdio>
#include <algorithm>
using namespace std;
int n,h1,m1,h2,m2,ans=1,temp;

struct t_time{
    int start;
    int finish;
}meets[1002];

bool cmp(t_time x,t_time y){
    if(x.finish==y.finish){
        return x.start<y.start;
    }
    return x.finish<y.finish;
}

int main(){
    scanf("%d",&n);
    for(int i=0;i<n;i++){
        scanf("%d:%d %d:%d",&h1,&m1,&h2,&m2);
        meets[i].start = h1*60 + m1;
        meets[i].finish = h2*60 + m2;
    }
    sort(meets, meets+n, cmp);
    temp = meets[0].finish;
    for(int i=1;i<n;i++){
        if(meets[i].start >= temp){
            ans++;
            temp = meets[i].finish;
        }
    }
    printf("%d",ans);
}
```

SUBMISSIONS

Dengan sedikit modifikasi pada IO, didapatkan memory yang lebih kecil dari yang semula **2.04 MiB** menjadi **0.52 MiB**.

Problem

[The Directors' Visitors](#)

Submitted

7 hours ago

Programming Language

C++ 11 (gnu 10.2)

Author

[Fajar\\_Satria\\_05111940000083](#)

100%

1 ms

0.52 MiB