

NAMA : Fajar Satria
NRP : 05111940000083
Kelas : PAA B

PERMASALAHAN 1 – HMLIS

The task is simply to find LIS and number of way we can select distinct LIS.

SOLUSI

Dengan berbekal literasi dari wikipedia yang berjudul [Longest Increasing Subsequence](#). Didapatkan solusi yang paling optimal untuk mencari LIS adalah dengan menggunakan Binary Search. Binary search tersebut berfungsi untuk mencari index posisi angka terakhir yang lebih kecil dari bilangan yang dicari didalam deret input. Deret input tersebut akan disimpan secara unik menaik dalam deret baru.

Deret input	batas atas	bilangan yg dicari	hasil Binary Search	Deret unik menaik
{1}	0	1	0	{1}
{1, 2}	1	2	1	{1, 2}
{1, 2, 3}	2	3	2	{1, 2, 3}
{1, 2, 3, 3}	3	3	2	{1, 2, 3}
{1, 2, 3, 3, 4}	4	4	3	{1, 2, 3, 4}
{1, 2, 3, 3, 4, 2}	5	2	1	{1, 2, 3, 4}
dst...				

Hasil dari binary search tersebut digunakan sebagai pembanding dengan ukuran dari deret unik yang dibuat selanjutnya. Jika hasilnya sama dengan ukuran dari deret unik, maka deret unik akan di perbesar dan diisi dengan bilangan yg dicari. Dengan demikian, deret input yang semula tidak unik dapat menjadi deret unik.

Setelah selesai dengan pencarian LIS, sekarang masuk ke tahap penghitungan count LIS. Untuk dapat menghitung jumlah dari LIS tersebut, maka harus disimpan semua angka terakhir / terbesar dari tiap iterasi Binary Search sebelumnya beserta jumlah sequence yang dapat dibuat dengan angka terakhir tersebut.

Deret input	LIS	Angka terakhir dari LIS	Deret baru yang dapat dibuat	Jumlah deret baru
{1}	1	{1}	{1}	1
{1, 2}	2	{2}	{1, 2}	1
{1, 2, 3}	3	{3}	{1, 2, 3}	1
{1, 2, 3, 3}	3	{3, 3}	{1, 2, 3}, {1, 2, 3}	2
{1, 2, 3, 3, 4}	4	{4}	{1, 2, 3, 4}, {1, 2, 3, 4}	2
{1, 2, 3, 3, 4, 2}	4	{4}	{1, 2, 3, 4}, {1, 2, 3, 4}	2
dst...				

Tahap terakhir adalah menggabungkan kedua potongan solusi diatas.

PSEUDOCODE

SOLVE(n)

1. size_lis = 0
2. let lis[1.. 100001] be a new array
3. let countlis[1.. 100001] be a new array
4. let history[1..100001][1...100001][2] be a new 3D array
5. for i=0 to n-1
6. read num
7. low = 0
8. high = size_lis
9. while (low < high)
10. mid = low + (high - low) / 2

```

11.         if num <= lis[mid]
12.             high = mid
13.         else    low = mid + 1
14.
15.     count = 1
16.     if low >= 1
17.         count = 0;
18.         for i=0 to size of history[low-1] - 1
19.             if history[low-1][i][0] < num
20.                 count = count + history[low-1][i][1] % mod
21.     history[low].push_back({ num, count})
22.
23.     if low = size
24.         lis[size]=num
25.         size = size + 1
26.     else    lis[low] = num
27.     countlis[low] = countlis[low] + count
28.
29.     if low>maks
30.         maks=low
31.
32. print ((maks + 1) % mod)
33. print (countlis[maks] % mod)

```

SOURCECODE

```

#include <cstdio>
#include <vector>
using namespace std;
const long long mod = 1000000007;
int n;
long long num,maks=0LL;
long long lis[100001];
long long countlis[100001];
vector<vector<pair<long long, long long>>> history(100001);

template<typename T>
T getNum(){
    T res=0;
    char c;
    while(1){
        c = getchar_unlocked();
        if(c==' '||c=='\n') continue;
        else break;
    }
    res=c-'0';
    while(1){
        c = getchar_unlocked();
        if(c>='0' && c<='9') res=10*res+c-'0';
        else break;
    }
    return res;
}
int main(){
    n=getNum<int>();
    long long size=0;
    while(n--){
        num=getNum<long long>();
        long long mid;
        long long low = 0;
        long long high = size;
        while(low < high){

```

```

        mid = low + (high - low) / 2;
        if(num <= lis[mid]){
            high = mid;
        }else{
            low = mid + 1;
        }
    }
    long long count = 1LL;
    if(low >= 1){
        count = 0;
        for(long long i=0;i<history[low-1].size();i++)
            if(history[low-1][i].first < num)
                count += history[low-1][i].second%mod;
    }
    history[low].push_back({num, count});

    if(low == size){
        lis[size++] = num;
    }else{
        lis[low] = num;
        countlis[low] += count;
        if(low>maks) maks=low;
    }
    printf("%lld %lld\n", (maks+1)%mod, countlis[maks]%mod);
}

```

SUBMISSIONS

Fajar: submissions
HowManyLis

ID		DATE	PROBLEM	RESULT	TIME	MEM	LANG
27896186		2021-05-16 13:36:59	HowManyLis	accepted edit ideone it	0.25	7.0M	CPP14

PERMASALAHAN 2 – Розклад

Нещодавно Козак Вус заснував власну компанію. Компанія дуже стрімко розвивається, а тому вже має багато працівників.

Козак Вус поставив своїм працівникам n задач. i -та задача має два параметри: r_i та c_i . r_i — момент часу, коли i -та задача повинна бути виконаною. c_i — важливість задачі (чим більше c_i , тим більш важлива задача).

Також Козак Вус задав деяку цілу сталу k .

Працівникам треба знайти такий масив e з n невід'ємних чисел, щоб наступне число було мінімально можливим:

$$\sum_{i=1}^n |r_i - e_i| \cdot c_i + \max(e) \cdot k =$$
$$= |r_1 - e_1| \cdot c_1 + |r_2 - e_2| \cdot c_2 + \dots + |r_n - e_n| \cdot c_n + \max(e) \cdot k$$

$\max(e)$ — максимальне число масиву e .

Козака Вуса не цікавить сам масив e . Він хоче дізнатися зазначене вище мінімальне можливе значення.

Допоможіть працівникам Козака Вуса розв'язувати цю задачу.

SOLUSI

Solusi yang saya gunakan adalah dengan merekonstruksi array e . Dikarenakan harga mutlak pada $|r_i - e_i|$ maka nilai e_i sudah pasti tidak melebihi r_i . Sehingga sebagai permulaan, array e adalah sama dengan array r .

Nilai minimum dapat dicari dari rumus yang diberikan. Dari rumus tersebut dapat dibagi menjadi 2 bagian, yaitu bagian kiri dan bagian kanan. Bagian kiri adalah sigma, dan bagian kanan adalah $\max(e) \cdot k$.

Dengan mengiterasi tiap nilai e_i tertinggi dari $e_i = r_i$ hingga $e_i = 0$ dengan menggunakan Binary Search. Maka nilai minimum dapat ditemukan seperti berikut.

k	r1	r2	r3	c1	c2	c3	e1	e2	e3	sum_left	sum_right	sum_total
1	1	2	3	1	2	3	1	2	3	0	3	3
1	1	2	3	1	2	3	1	2	2	3	2	5
1	1	2	3	1	2	3	1	1	2	5	2	7
1	1	2	3	1	2	3	1	1	1	8	1	9
1	1	2	3	1	2	3	0	1	1	9	1	10
1	1	2	3	1	2	3	0	0	1	11	1	12
1	1	2	3	1	2	3	0	0	0	14	0	14
										minimum		3

Terdapat beberapa basecase sehingga tidak perlu melalui beberapa tahapan diatas, salah satunya adalah ketika $c_i=0$. Maka sudah pasti hasilnya juga akan 0. Seperti tertera dalam tabel berikut.

k	r1	r2	r3	c1	c2	c3	e1	e2	e3	sum_left	sum_right	sum_total
1	1	2	3	0	0	0	1	2	3	0	3	3
1	1	2	3	0	0	0	1	2	2	0	2	2
1	1	2	3	0	0	0	1	1	2	0	2	2
1	1	2	3	0	0	0	1	1	1	0	1	1
1	1	2	3	0	0	0	0	1	1	0	1	1
1	1	2	3	0	0	0	0	0	1	0	1	1
1	1	2	3	0	0	0	0	0	0	0	0	0
										minimum		0

Untuk basecase lain, saya belum bisa mendapatkan solusi yang tepat.

PSEUDOCODE

SOLVE(R, C, k)

```
1. max_ = 0
2. last = 0
3. n = R.length
4. let E[1..n] be new arrays
5. for i=0 to n-1
6.     if R[i] > max_
7.         max_ = R[i]
8.     E[i] = R[i]
9. for i=0 to n-1
10.    if C[i] > 0
11.        last = last + R[i]*C[i]
12.    if R[i] > max_
13.        max_ = R[i]
14.
15. if last = 0
16.     return 0
17.
18. res = last+max_*k
19. while max_ > 0
20.     low = 0
21.     high = n-1
22.     while low < high
23.         mid = low + (high - low) / 2
24.         if max_ <= E[mid]
25.             high = mid;
26.         else low = mid + 1
27.     e[low] = e[low] - 1
28.     max_ = max - 1
29.     now = last-(R[low]-(E[low]+1))*C[low]+(R[low]-E[low])*C[low]
30.     for i=low to n-1
31.         if E[i] > max_
32.             max_ = E[i]
33.     res = min(res, now+max_*k)
34.     last = now
35. return res
```

SOURCECODE

```
#include <cstdio>
#include <algorithm>
using namespace std;
unsigned long long last=0LLU,now=0LLU,res=0LLU,sum_r=0LLU;
template<typename T>
T getNum() {
    T res=0;
    char c;
    while(1) {
        c = getchar_unlocked();
        if(c==' ' || c=='\n') continue;
        else break;
    }
    res=c-'0';
    while(1) {
        c = getchar_unlocked();
        if(c>='0' && c<='9') res=10*res+c-'0';
```

```

        else break;
    }
    return res;
}
int main(){
    int n,k;
    n=getNum<int>();
    k=getNum<int>();
    int r[n],c[n],e[n];
    int low,mid,high,max_=0,max_c=0;
    for(int i=0;i<n;i++){
        r[i]=getNum<int>();
        e[i]=r[i];
    }
    for(int i=0;i<n;i++){
        c[i]=getNum<int>();
        if(c[i]==0) continue;
        else if(c[i]==1) sum_r += r[i];
        last += r[i]*c[i];
        if(r[i]>max_) max_=r[i];
        if(c[i]>max_c) max_c=c[i];
    }
    if(max_c==0){
        puts("0");
        return 0;
    }else if(max_c==1){
        printf("%llu\n",min(sum_r,1LLU*max_*k));
        return 0;
    }
    res = min(last,1LLU*max_*k);
    last=0LL;
    while(max_>0){
        low = 0;
        high = n-1;
        while(low < high){
            mid = low + (high - low) / 2;
            if(max_ <= e[mid]){
                high = mid;
            }else{
                low = mid + 1;
            }
        }
        e[low]--;
        max_--;
        now = last-(r[low]-(e[low]+1))*c[low]+(r[low]-e[low])*c[low];
        for(int i=low;i<n;i++){
            if(e[i]>max_) max_=e[i];
        }
        res = min(res,now+max_*k);
        last = now;
    }
    printf("%llu\n",res);
}

```

SUBMISSIONS

Gmail - Tulis Surat

Solution #9004535 - Розклад

← → ↺ 🏠 🔒

https://www.e-olymp.com/en/submissions/9004535

📄 ⋮ 📌 ⭐ 🔍 Search

🔴 5 >> ☰

Problem

Submitted

Programming Language

Author

Розклад

1 hour ago

C++ 17 (gnu 10.2)

Fajar_Satria_05111940000083

8%

0 ms

0 MiB

Test #	Status	Score	Duration	CPU	Memory
✓ Test suite #0	Accepted	0 / 0	1 ms	1 ms	72 KiB
✓ Test #1	Accepted	0 / 0	1 ms	1 ms	72 KiB
✓ Test #2	Accepted	0 / 0	1 ms	1 ms	68 KiB
✓ Test #3	Accepted	0 / 0	1 ms	1 ms	72 KiB
✓ Test suite #1	Accepted	8 / 8	39 ms	39 ms	11,756 KiB
✗ Test suite #2	Wrong answer	0 / 15	39 ms	38 ms	11,760 KiB