



第七屆高一排名賽 題解

關於這次比賽

1.

撲克

First AC: 陳凱昕

AC: 22 人

出題者: 李長諺

子任務1

判斷各種花色下一個是甚麼
數字直接輸出就行

子任務2

特判黑桃

輸出下個數字

2.

重組回文

First AC: 方歆閔

AC: 11 人

出題者: 周暘典

題目

給定一個字串。

複製一段前綴，重組後貼到字串後面，使得最後是一段回文。

子題1: 所有字元皆不相同

皆不相同?

ABCDE 複製ABC \Rightarrow ABCDE + [ABC]重組

無論怎麼重組, 組起來都不是回文(因為一定不同)。

複製整段? ABCDE 只需要 ABCDEDCBA 即可。

→ 輸出 N-1

子題2: $N \leq 8$

列舉複製了哪一段。

AABC 可以複製 A, AA, AAB, AABC
共 N 段。

任意重組 $\rightarrow O(N!)$

檢查是不是回文 $\rightarrow O(N)$

總複雜度 $O(N^2 \cdot N!) = O((N+2)!)$

子題3: 存在某一個 j 符合 $a_1 \sim a_j$ 相等
， $a_j \neq a_{j+1}$ ，且 $a_{j+1} \sim a_n$ 也相同

舉例: AAABB

複製任何一段 $AAABB + [A..AB..B]$

最後回文的樣子? AAABB..BA...A

→ 不必複製B, A的數量等於前綴數量即可

→ 找到 j 並輸出

子題4: 存在某一個 j 符合 $a_i \neq a_{j+1}$ ($1 \leq i \leq j$), 且 $a_{j+1} \sim a_n$ 相同

舉例: CDCBBB

比對時, 複製的那一段不可以對到B。

→ 只能複製到前面那一段

怎麼重組呢?

→ 依長度來看, 會對到複製的那一段...



重組成 反過來的前綴

子題4: 存在某一個 j 符合 $a_i \neq a_{j+1}$ ($1 \leq i \leq j$), 且 $a_{j+1} \sim a_n$ 相同

→ 找到 j 並輸出

題目範圍

依照比對的長度來看，複製的前綴會對到原本的前綴。

依照想法，複製出的那一段可以和前綴相同。
也就是說剩下的部分要是回文。

AAABAABA => AAABAABAAA

→ 找到從字尾結束最長的回文

題目範圍

找到從字尾結束最長的回文

列舉開頭, 檢查開頭到字尾是不是回文。

列舉 $O(N)$, 每次檢查 $O(N)$

總共 $O(N^2)$

3.

賭博

First AC: 歐育淇

AC: 3 人

出題者: 成俊宏

YAAAAAAAAA 數論題~~~~~

YAAAAAAAAA 數論題~~~~~

YAAAAAAAAA 數論題~~~~~

YAAAAAAAAA 數論題~~~~~

YAAAAAAAAA 數論題~~~~~

YAAAAAAAAA 數論題~~~~~

YAAAAA 數論題~~~~~

YAAAAA 數論題~~~~~

YAAAAA AAAA 數論題~~~~~

YAAAAAAAAA 數論題~~~~~

YAAAAA 數論題~~~~~

YAAAAAAAAA 數論題~~~~~

沒 我們來看子任務

必定符合 $-10^{100} \leq A \leq 10^{100}$ ， $-10^{16} \leq B \leq 10^{16}$ 。

| 編號 | 分數 | 子任務內容 |
|----|------|---|
| 1 | 5 % | $A = 0$ 或 $B = 1$ |
| 2 | 6 % | $-10^9 \leq A \leq 10^9$ ， $-10^8 \leq B \leq 10^8$ |
| 3 | 7 % | $-10^{18} \leq A \leq 10^{18}$ |
| 4 | 27 % | $A, B \geq 0$ |
| 5 | 23 % | $B = \{2, 3, 4, 5, 6, 8, 9\}$ 其中之一 |
| 6 | 32 % | 題目範圍 |

1號子任務

A一定是B的倍數
可以特判掉 (?!?)

```
cin >> A >> B;  
cout << "Yes" << endl;
```

2, 3號子任務

A B 開int 解2號子任務

開long long 解3號

直接%% %爆

```
int A, B;  
cin >> A >> B;  
if (A%B) cout << "No" << endl;  
else cout << "Yes" << endl;
```

```
long long A, B;  
cin >> A >> B;  
if (A%B) cout << "No" << endl;  
else cout << "Yes" << endl;
```

4號子任務

其實這個要寫出來

差不多就是正解了

我們留到正解解釋OuOb

5號子任務

倍數判別法

但還卡了一個好玩的東西

直接寫還是會WA

後面解釋


```
string A;
int B;
cin >> A >> B;
if (B == 2 or B == 4 or B == 8) {
    int tmp = 0;
    for (int i=A.size()-3;i<A.size();i++) tmp = tmp*10 + (int)(A[i]-'0');
    if (tmp%B) cout << "No" << endl;
    else cout << "Yes" << endl;
}
if (B == 3 or B == 9) {
    int tmp = 0;
    for (int i=(A[0]=='-');i<A.size();i++) tmp += (int)(A[i]-'0');
    if (tmp%B) cout << "No" << endl;
    else cout << "Yes" << endl;
}
if (B == 5) {
    if (A[A.size()-1] == '0' or A[A.size()-1] == '5') cout << "Yes" << endl;
    else cout << "No" << endl;
}
if (B == 6) {
    int tmp = 0;
    for (int i=(A[0]=='-');i<A.size();i++) tmp += (int)(A[i]-'0');
    if (tmp%3==0 and (A[A.size()-1]%2==0)) cout << "Yes" << endl;
    else cout << "No" << endl;
}
```

正解時間

正解

A是一個大數 不能用int或long long 處理

B 則是能用long long 存

正解是用同餘定理~~~

123456789

```
string A;  
long long B;  
  
cin >> A >> B;  
  
int a = 0;  
for (int i = (A[0] == '-'); i < A.size(); i++) {  
    a = a * 10 + (A[i] - '0'); a %= B;  
}  
if (a) cout << "No" << endl;  
else cout << "Yes" << endl;
```

4.

停止爆炸

First AC: 曹宸睿

AC: 4 人

出題者: 周暘典

題目

給定 $N \times M$ 的格子和 Creeper 的位置、Player 的位置, 問能不能炸到 Player。

子題1: $A_{ij} = 0$

完全沒有阻礙。

輸出 “yes” !!

子題2: $N = 1$

苦力怕和 Player 在同一條線上。

苦力怕會往Player的方向走。

→ 迴圈檢查Creeper能不能走到範圍內

子題3: $A_{ij} = 0$ or 3

[3 到 0 / 0 到 3] 皆不可行走。

→ 0只能走到0, 3只能走到3

→ DFS 檢查能否連通到範圍內。

子題4: $1 \leq N, M \leq 5$

當然本題是要爆搜。

當你忘記標 vis 的時候:

$O(4^D)$ 最差 $O(4^{NM})$

因為步數沒那麼多, 所以其實跑得完。

題目範圍

爆搜。

記得標 vis 阿!!!

複雜度: $O(NM)$

5.

挖礦小隊

First AC: 蔡承翰

AC: 5 人

出題者: 李長諺

來看每個子任務吧

子任務表

必定符合 $1 \leq N \leq 200,000$ $0 \leq k \leq 10^{12}$ $0 \leq a_i \leq 10^9$

| 編號 | 分數 | |
|----|------|--|
| 1 | 13 % | $k > (\text{最大的 } a_i) - (\text{最小的 } a_i)$ 或者 $k = 0$ |
| 2 | 15 % | $a_{i-1} \leq a_i, 1 \leq i \leq N \leq 10,000$ |
| 3 | 31 % | $a_{i-1} \leq a_i, 1 \leq i \leq N \leq 200,000$ |
| 4 | 41 % | 題目範圍 |

子任務1

k 比最大跟最小的差還大 \rightarrow 取全部的人

$k=0$ \rightarrow 只能取一樣大的(不一定都是取最大的)

```

if(k==0)
{
    sort(a,a+n);
    long long ans=a[0];
    long long sum=a[0];
    for(long long i=1;i<n;i++)
    {
        if(a[i]>a[i-1])
        {
            ans=max(ans,sum);
            sum=a[i];
        }
        else
        {
            sum+=a[i];
        }
    }
    ans=max(ans,sum);
    cout<<ans<<endl;
    return 0;
}

```

其他

```

long long sum=0;
for(long long i=0;i<n;i++)
{
    sum+=a[i];
}

```

子任務2

ai遞增, $n < 10000$

$O(n^2)$

計算從每個頭開始找

找到不能再往後找為止

值得注意的是:

假如目前區間 $[L, R]$

最小就是L最大就是R

只要 $R - L \leq k$ 就是合法的

```
long long ans=0;
for(long long i=0;i<n;i++)
{
    long long sum=0;
    for(long long j=i;j<n;j++)
    {
        if(a[j]-a[i]>k)break;
        sum+=a[j];
    }
    ans=max(ans,sum);
}
cout<<ans<<endl;
```


子任務3

小性質:

假如 $[i,R]$ 是合法的

$[i+1,R]$ 必然也會合法 ($R>i$)

所以可以利用上個子任務時

往後找的時候可以儲存上次的結果

利用到這次

```
long long p=0;
long long ans=0;
long long sum=a[0];
for(long long i=0;i<n;i++)
{
    if(p<i)
    {
        p=i;
        sum=a[i];
    }
    while(p+1<n&& a[p+1]-a[i]<=k)
    {
        p++;
        sum+=a[p];
    }
    ans=max(ans,sum);
    sum-=a[i];
}
cout<<ans<<endl;
```

子任務4

sort之後就跟子任務3一樣了

來看看有人100分的code

```
19     while(sp[a]-sp[head]>k)
20     {
21         ans-=sp[head];
22         head++;
23
24     }
25     ans+=sp[a];
26     //cout<<ans<<" ";
27     aans=max(aans,ans);
```

也可以從尾巴來看
然後head也是遞增所以 $O(n)$

```
sort(arr+1, arr+1+n);  
for ( int i=1; i<=n; ++i )    pre[i] = pre[i-1]+arr[i];  
for ( int i=1; i<=n; ++i ) {  
    int pos = upper_bound(arr+1, arr+1+n, arr[i]+k)-arr-1;  
    ans = max(ans, pre[pos]-pre[i-1]);  
}
```

來看看歐先生的作法

```
4 long long aaa[100020];
5
6 int main() {
7     ios::sync_with_stdio(0);
8     cin.tie(0);
9
10    long long n,i,a,ans=0,k,ouo=0;
11    cin>>n>>k;
12
13    for(i=0;i<n;i++)
14    {
15        cin>>a;
16        aaa[a]+=a;
17        ouo=max(ouo,a);
18    }
19
20    for(i=1;i<=100000;i++)
21    {
22        aaa[i]+=aaa[i-1];
23    }
24    for(i=100000;i>k;i--)
25    {
26        aaa[i]-=aaa[i-k-1];
27    }
28    for(i=0;i<=100000;i++)
29    {
30        ouo=max(ouo,aaa[i]);
31    }
32
33    cout<<ouo<<"\n";
34}
```

6.

狗狗大隊

First AC: 曹宸睿

AC: 3 人

出題者: 周暘典

題目

給定一個序列。

找出多段長度為 K 的連續自序列, 使他們的總和最大。

兩段之間不可以相鄰。

子題1: $2K > N$

意思是，找不出不重疊的2段。

找出長度為 K 的序列的最大總和。

用[前綴和 / 資料結構 / 爬行法]可以完成。

子題2: $a_{i-1} \leq a_i$

後面的比較大?

選後面的一定比較好(證明)。

利用迴圈可以解決,也可以使用前綴和。

子題3: $1 \leq K \leq N \leq 7$

爆搜, 列舉每一段, 並計算最大值。

子題4: $1 \leq N \leq 100,000$, $K = 1$

DP。

若 $dp(n)$ 代表前 n 項最大,

可得 $dp(n) = \max(dp(n-2)+v[n], dp(n-1))$

題目範圍

DP。

若 $dp(n)$ 代表前 n 項最大,

$S(a,b)$ 代表 $a\sim b$ 項的合。

可得 $dp(n) = \max(dp(n-k-1)+S(n-k+1, n), dp(n-1))$, S 可以用前綴和處理。

7.

246 三硝基甲苯

First AC: 歐育淇

AC: 5 人

出題者: 陳冠辰

題目

給定一個序列，以及一些操作，問最後序列的樣貌。

操作：每次把一段前綴改成為一個數字。

子任務1: $1 \leq N, Q \leq 5,000$

暴力做。

複雜度 $O(NQ)$

子任務2: $B_{i-1} \leq B_i$

B_i 會把 B_{i-1} 蓋掉。

把最後一次的操作暴力作即可。

子任務3: $1 \leq N, Q \leq 300,000$

試試看把序列切成 **根號N** 段。

每次暴力更新，利用一個序列來記錄更新的狀態。

複雜度 $O(QN^{2/3})$

子任務4: $B_{i-1} \geq B_i$

每一次的更新都越來越靠左邊。

從後面開始更新，每一次紀錄目前更新到的顏色。

題目範圍

紀錄每一項最後被更新到的時間點和值。

因為最右邊的不會被覆蓋，可以從右邊開始更新，每一次檢查目前的時間點是否大於目前項的時間點並更新。

8.

下水道

First AC: 歐育淇

AC: 1 人

出題者: 周暘典

題目

給定一個長度 N 的序列，其中有 M 項有阻塞。
試著將整段序列清空。

子題1: $N \leq 300$

一個一個問，問到了再更新。

用到 $300+M$ 次，剛剛好。

子題2: $N \leq 1,000$

把序列切成10個一段。

一個一個問每一段，並將含有阻塞的(不到10段)暴力詢問。

總共最多 $100 + 100 + M = 200 + M$ 次

子題3: $N \leq 10,000$, $M \leq 2$

把序列切成100個一段。

一個一個問每一段，並將含有阻塞的(不到2段)
暴力詢問。

總共最多 $100 + 200 + M = 300 + M$ 次

子題4: $N \leq 1,000,000$, $M = 1$

試著二分。

每一次看看左邊有沒有，再看看右邊有沒有。

持續往有的那邊走，最終一定會找到那一塊。

總共最多 $\log_2 1000000 + 1 \leq 21$ 次

題目範圍

兩邊只能問其中一邊。

試著每次找出最左邊的並清除，最後一定能清除整個序列。

最多 $10 \cdot \log_2 1000000 + M = 200 + M$ 次。