

第六屆高一生程式設計排名賽

解題報告

A.好狗狗

命題:lys0829

解題狀況

AC: 18人

First Blood: 黃宥嘉 (00:03:43)

A.好狗狗--題目大意

給你一個數字，要你判斷是不是每一位數都是質數

e.x 2357

A.好狗狗

n最大可以是 10^{100}

我記得long long int好像只能存到 $2^{63}-1$

這麼大要怎麼存阿!!

A.好狗狗

大家應該都學過**string**!!!

沒錯, 你猜對了, 就是用*string*

小於10的質數只有**2,3,5,7**

判斷一下是不是每位都是這幾個數字就好了

A.好狗狗

```
1  #include<bits/stdc++.h>
2  using namespace std;
3  bool isprime(char a){
4      if(a=='2' || a=='3' || a=='5' || a=='7'){
5          return true;
6      }
7      return false;
8  }
9  int main(){
10     string s;
11     cin>>s;
12     bool ch=true;
13     int ss=s.size();
14     for(int i=0;i<ss && ch;i++){
15         if(!isprime(s[i])){
16             ch=false;
17         }
18     }
19     if(ch){
20         cout<<"Yes"<<endl;
21     }
22     else{
23         cout<<"No"<<endl;
24     }
25     return 0;
26 }
```

短短的code, 輕鬆搞定

B.樓梯間

命題:Ice1187

解題狀況

AC: 11人

First Blood: 黃宥嘉 (00:12:53)

B.樓梯間 題目大意

給一個 $n*m$ 的陣列, 算所有元 $(2i-2j)$ 的和 mod p 。

B.樓梯間

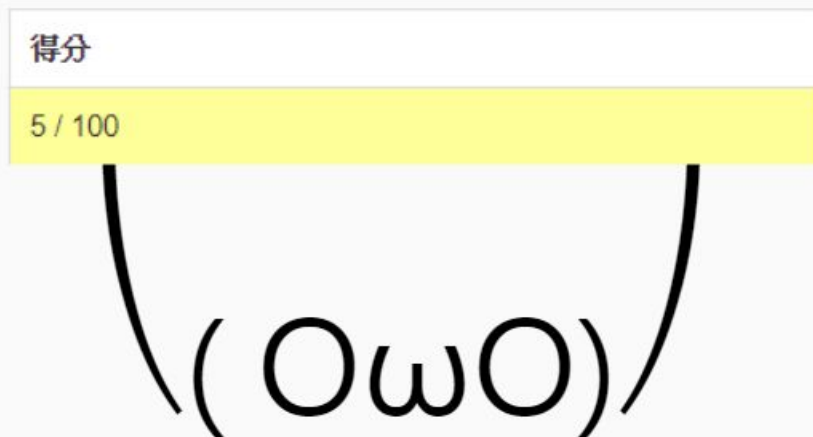
```
int main(){
    ios::sync_with_stdio(0);
    cin.tie(0);
    //
    int n, m, p, ans = 0;
    cin >> n >> m >> p;
    for(int i=1; i<=n; i++){
        for(int j=1; j<=m; j++){
            ans += 2*(i-j);
        }
    }

    cout << ans << '\n';
    return 0;
}
```

直接暴力啊！！！！

(OωO)

5 points get ! ! !



B.樓梯間

```
cout << ans%p << '\n';  
return 0;
```

這次記得mod p

$$/(//\cdot\omega//)/$$

B.樓梯間

Stair

▶ Scored (18.0 / 100.0)

變18分了 (" ∇ ")

B.樓梯間

提示

$9 + 9 + 9 + 9 + 9 + 9 + 9 + 9 + 9 = 9 * 9$ ，這不是文具店業配。



三角形面積公式

$$2i - 2j = 2(i - j)$$

$$\Rightarrow 2(mi - \frac{m(m+1)}{2})$$

B.樓梯間

餘數定理

$$n = ap + r$$

$$m = bp + R$$

$$n - m = (a - b)p + (r - R)$$

$$n \% p = r$$

$$m \% p = R$$

$$(n - m) \% p = r - R$$

$$\Rightarrow nm(n - m) \% p = (n \% p)(m \% p)((n - m) \% p) \% p = rR(r - R)$$

```
int main(){
    ios::sync_with_stdio(0);
    cin.tie(0);
    //
    long long n, m, p, ans = 0, mmod;
    cin >> n >> m >> p;
    mmod = m%p;
    for(int i=1; i<=n; i++){
        ans += (2*(mmod*(i%p) - mmod*((m+1)%p)/2))%p;
        ans %= p;
    }
    cout << ans%p << '\n';
    return 0;
}
```

B.樓梯間

Stair

▶ Scored (35.0 / 100.0)

35 points www

B.樓梯間

提示

$9 + 9 + 9 + 9 + 9 + 9 + 9 + 9 + 9 = 9 * 9$ ，這不是文具店業配。



B.樓梯間

$$(\lceil \cdot \omega \cdot \rceil)$$
$$\Sigma$$

數學老師上課教過的sigma, 還有人記得嗎? (根本都在睡覺#

B.樓梯間 **正解**

$$\backslash(\nabla^{\circ})\wedge(^{\circ}\nabla^{\circ})\wedge(^{\circ}\nabla)\wedge(\nabla^{\circ})\wedge(^{\circ}\nabla^{\circ})\wedge(^{\circ}\nabla)/$$

$$[\sum_{i=1}^n \sum_{j=1}^m (2 \times i - 2 \times j)] \mod p = [m \times n \times (n - m)] \mod p$$

$$\backslash(\nabla^{\circ})\wedge(^{\circ}\nabla^{\circ})\wedge(^{\circ}\nabla)\wedge(\nabla^{\circ})\wedge(^{\circ}\nabla^{\circ})\wedge(^{\circ}\nabla)/$$

B.樓梯間

```
int main() {
    long long n, m, p;
    long long mod, nod;
    long long ans;
    cin >> n >> m >> p; //  $\sigma(i=1, n)\sigma(j=1, m)(2i-2j)$ 
    mod = m%p;
    nod = n%p;
    ans = (((mod)*(nod))%p) * ((n-m)%p) %p ; //  $m*n*(n-m)$ 
    cout << ans << '\n';
    return 0;
}
```

B.樓梯間

Stair | ▶ Scored (100.0 / 100.0) |

AC了呢, ㄅㄅ :3

B.樓梯間

1. 暴力解, 不用 $\text{mod } p$
2. 暴力解, 要 $\text{mod } p$
3. 單層暴力, 要 $\text{mod } p$
4. 正解

C.藏寶圖

命題:MiohitoKiri5474

解題狀況

AC: 0人 QwQ

First Blood: NULL (00:00:00) QwQ

Highest Score : 23pts by. 黃宥嘉

首先呢，我們先來看看題目要做什麼

題目有提到，小黑是路癡，為了避免一直繞圈圈

又，不希望浪費太多腦力，所以希望圖上所有路徑總和最小

啊那不就是MST嗎

P.S. MST, 全名為Minimum Spanning Tree, 中文正式名稱為最小生成樹

說到MST

有兩種做法，我們這邊用 Kruskal

Kruskal

先依照邊權重把所有邊排序, 接著按照順序對所有邊檢查當前邊之兩端點是否屬於同一個聯通塊中

- > 同一個聯通塊

- >> 跳過

- > 不同的聯通塊

- >> 加進去MST中

說到排序

寫個cmp或operator吧

```
1  #include<bits/stdc++.h>
2
3  using namespace std;
4
5  struct node{
6      int u, v, w;
7  };
8
9  // cmp
10 inline bool cmp ( node a, node b ){
11     return a.w < b.w;
12 }
13
14 // operator
15 inline bool operator < ( node a, node b ){
16     return a.w < b.w;
17 }
```

差不多像這樣

說到檢查聯通塊

那就寫個 Disjion Set 吧, 順便寫個路徑壓縮壓複雜度

```
1  #define maxN 1000005
2
3  int dis[maxN];
4
5  inline void init ( void ){
6      for ( int i = 0 ; i < maxN ; i++ )
7          dis[i] = i;
8  }
9
10 int find ( int n ){
11     return dis[n] == n ? n : dis[n] = find ( dis[n] );
12 }
13
14 inline void Union ( int a, int b ){
15     dis[find ( a )] = find ( b );
16 }
17
18 inline bool same ( int a, int b ){
19     return find ( a ) == find ( b );
20 }
```


沒意外的話Kruskal大概長這樣
(請搭配前面的code服用)

```
1  typedef pair < int, int > pii;  
2  
3  vector < node > edges;  
4  vector < pii > mst;  
5  
6  inline void Kruskal ( void ){  
7      sort ( edges.begin(), edges.end(), cmp );  
8      for ( auto i: edges ){  
9          if ( same ( i.u, i.v ) )  
10             continue;  
11             Union ( i.u, i.v );  
12             mst[i.u].push_back ( pii ( i.v, i.w ) );  
13             mst[i.v].push_back ( pii ( i.u, i.w ) );  
14         }  
15     }
```

優西，那麼我們來處理樹上最遠點對吧

可能另外一個名詞比較常出現，我們稱為樹直徑

『~~不管是什麼題目，都先寫寫暴力法！~~』

我們先寫寫暴力 dfs N 次的做法(搞不好就不小心 AC 了呢)

沒意外的話，應該會長這樣 ->

```
1  int dist[maxN];
2  bool visited[maxN];
3  void dfs ( int n ){
4      visited[n] = true;
5      for ( pii i: mst[n] ){
6          if ( visited[i.F] )
7              continue;
8          dist[i.F] = dist[n] + i.S;
9          dfs ( i.F );
10     }
11 }
12 // in main function
13 int ma = -1;
14 for ( int i = 0 ; i < n ; i++ ){
15     memset ( dist, 0, sizeof dist );
16     memset ( visited, 0 , sizeof visited );
17     dfs ( i );
18     for ( int j = 0 ; j < n ; j++ )
19         ma = max ( ma, dist[j] );
20 }
21 cout << ma << '\n';
22
```

Subtask 1

(0 / 0)

Subtask 2

(16 / 16)

Subtask 3

(0 / 29)

#	Outcome	Details	Execution time	Memory used
7	Not correct	Execution timed out	4.948 s	44 MiB
8	Not correct	Execution timed out	4.966 s	44 MiB
9	Not correct	Execution timed out	4.977 s	44 MiB
10	Not correct	Execution timed out	4.983 s	44 MiB
11	Not correct	Execution timed out	5.000 s	44 MiB

恭喜拿到16分
後面直接吃TLE了w

Subtask 4

(0 / 23)

Subtask 5

(0 / 19)

我們來想想，有哪些性質是我們可以用的

首先，我們知道，如果某個點是樹直徑的兩端點之一，那麼這個點的度數 絕對為1

假設現在樹上有隨意一個點，並在樹直徑上

> 不在兩端

>> 度數絕對 ≥ 2 (至少要連接兩端)

> 假設 在兩端 且 度數 ≥ 2

>> 扣除掉連接往另一端點的一條邊後，還有另一條，代表樹直徑可以往另一邊繼續延伸

>>> 矛盾 (不在兩端點)

所以只要dfs度數 <2 的點就好啦

輕輕鬆鬆

事不宜遲趕快來 judge 看看吧w

沒意外的話code應該會長這樣

```
1  int dist[maxN];
2  bool visited[maxN];
3  void dfs ( int n ){
4      visited[n] = true;
5      for ( pii i: mst[n] ){
6          if ( visited[i.F] )
7              continue;
8          dist[i.F] = dist[now] + i.S;
9          dfs ( i.F );
10     }
11 }
12 // in main function
13 int ma = -1;
14 for ( int i = 0 ; i < n ; i++ ){
15     if ( mst[i].size() != 1 )
16         continue;
17     memset ( dist, 0, sizeof dist );
18     memset ( visited, 0 , sizeof visited );
19     dfs ( i );
20     for ( int j = 0 ; j < n ; j++ )
21         ma = max ( ma, dist[j] );
22 }
23 cout << ma << '\n';
```

如果是我就會去寫下一題子

➤ Subtask 1

(0 / 0)

➤ Subtask 2

(16 / 16)

➤ Subtask 3

(29 / 29)

▼ Subtask 4

(0 / 23)

#	Outcome	Details	Execution time	Memory used
12	Not correct	Execution timed out	4.988 s	40.1 MiB
13	Not correct	Execution timed out	4.971 s	40.1 MiB
14	Not correct	Execution timed out	4.975 s	40.1 MiB
15	Not correct	Execution timed out	4.994 s	40.1 MiB
16	Not correct	Execution timed out	4.976 s	40.1 MiB

➤ Subtask 5

(0 / 19)

恭喜多拿29分

再重新看一次code, 看看哪邊可以壓常數

Tips!

1. 對一棵樹 dfs 時不用紀錄這個點是否已經走過, 只要在 dfs 時順便丟進他的父節點, 並在 loop 時檢查是否為父節點就好了
2. 反正所有的數字點的距離都是從第一個節點開始累加, 所以只要把起始點 (root) 改成 0 就好了

不需要紀錄 visited, 我們把 visited 拔掉

於是 code 變成這樣惹

```
1  int dist[maxN];
2  void dfs ( int n, int p ){
3      for ( pii i: mst[n] ){
4          if ( i.F == p )
5              continue;
6          dist[i.F] = dist[n] + i.S;
7          dfs ( i.F );
8      }
9  }
10 // in main function
11 int ma = -1;
12 for ( int i = 0 ; i < n ; i++ ){
13     memset ( dist, 0, sizeof dist );
14     dfs ( i, -1 );
15     for ( int j = 0 ; j < n ; j++ )
16         ma = max ( ma, dist[j] );
17 }
18 cout << ma << '\n';
```

不需要每次都 memset, 把起始點改成 0

於是乎 code 又改動惹

```
1  int dist[maxN];
2  void dfs ( int n, int p ){
3      for ( pii i: mst[n] ){
4          if ( i.F == p )
5              continue;
6          dist[i.F] = dist[n] + i.S;
7          dfs ( i.F );
8      }
9  }
10 // in main function
11 int ma = -1;
12 for ( int i = 0 ; i < n ; i++ ){
13     dist[i] = 0
14     dfs ( i, -1 );
15     for ( int j = 0 ; j < n ; j++ )
16         ma = max ( ma, dist[j] );
17 }
18 cout << ma << '\n';
```

再 judge 一次吧。。。

恭喜，沒有多拿分數w

› Subtask 1

(0 / 0)

› Subtask 2

(16 / 16)

› Subtask 3

(29 / 29)

› Subtask 4

(0 / 23)

› Subtask 5

(0 / 19)

› Subtask 6

(0 / 13)

來想正解了啦~~(假解什麼的都是邪門歪道)~~

呃，其實求樹直徑不難

step 1. 隨便找一個點 dfs 下去，找到最遠點並記錄其編號

setp 2. 用那個點再 dfs 一次，找到的最遠距離即是答案

P.S. 證明請自己上網查，要講解太花費時間了~~(背起來，不然就用心去感受)~~

~~(明明就是不會是在牽拖什麼)~~

繼續 judge 吧(倒

直接dfs 兩次！

```
38 void dfs ( int n, int p ){
39     for ( auto i: mst[n] ){
40         if ( i.first == p )
41             continue;
42         dist[i.first] = dist[n] + i.second;
43         dfs ( i.first, n );
44     }
45 }
46 int main(){
47     ios::sync_with_stdio ( false );
48     cin.tie ( 0 );
49     int n, m, u, v, w, idx, ma = -1;
50     cin >> n >> m;
51     while ( m-- ){
52         cin >> u >> v >> w;
53         edges.push_back ( node { u, v, w } );
54     }
55     Kruskal();
56     dfs ( 0, -1 );
57     for ( int i = 0 ; i < n ; i++ )
58         if ( ma < dist[i] )
59             idx = i, ma = dist[i];
60     dist[idx] = 0;
61     dfs ( idx, -1 );
62     for ( int i = 0 ; i < n ; i++ )
63         ma = max ( ma, dist[i] );
64     cout << ma << '\n';
65 }
```

尷尬，怎麼只有 87 分(X

› Subtask 1

(0 / 0)

› Subtask 2

(16 / 16)

› Subtask 3

(29 / 29)

› Subtask 4

(23 / 23)

› Subtask 5

(19 / 19)

› Subtask 6

(0 / 13)

87分不能再高了w

阿, $N \leq 10^6$ (只開5次的等等請自由落體下樓)

> 最遠距離可能會爆int (有事沒事開long long!)

- $1 \leq N \leq 10^6$
- $1 \leq M \leq 3 * 10^6$ & $N - 1 \leq M$
- $0 \leq u, v < N$
- $w \leq 10^4$
- 保證給定的路徑有連通所有據點，亦保證藏寶圖唯一。

改完後終於AC惹 Orz

▼ Subtask 6

(13 / 13)

#	Outcome	Details	Execution time	Memory used
21	Correct	Output is correct	0.489 s	30.5 MiB
22	Correct	Output is correct	3.100 s	97 MiB
23	Correct	Output is correct	3.128 s	97 MiB
24	Correct	Output is correct	3.259 s	97 MiB
25	Correct	Output is correct	3.159 s	97 MiB

總結一下

0pts: 範測, 測試用(格式之類的)

11pts: 會MST就拿得到

25pts: 把MST權重和輸出就是答案

23pts: 會樹直徑 / 樹上最遠點對就可以了(不用寫MST)

19pts: 很單純的樹上最遠點對, 邊權重還保證都一樣

13pts: 不是正解過不了

突然想想
是不是很简单呢w

D. 普查

命題:Ice1187

解題狀況

AC: 4人

First Blood: 李長諺 (00:30:02)

D. 普查 題目大意

給一個長度 n 、編號從1開始的陣列，求 $[1, r]$ 區間最大。

直接暴力啊！！！！

\(O\omega O)\)

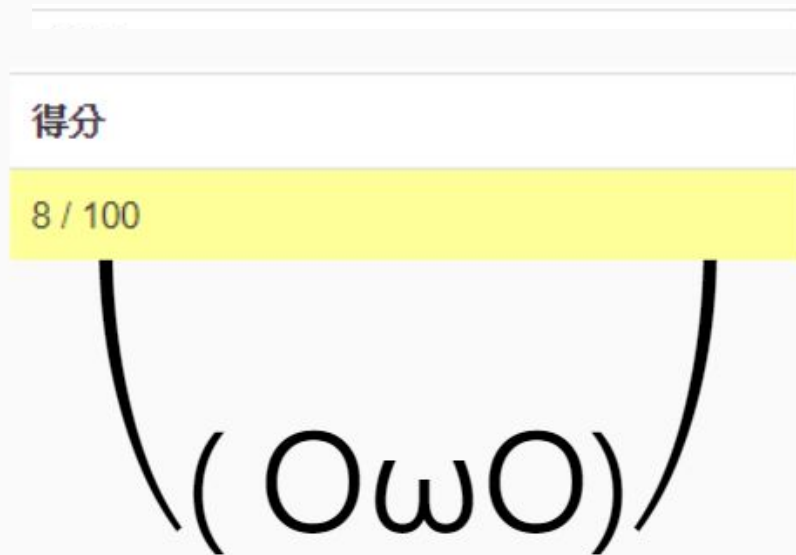
D. 普查 題目大意

```
Long Long arr[1000005];

int main(){
    ios::sync_with_stdio(0);
    cin.tie(0);
    // init
    int n, q, l, r;
    Long in, max;
    cin >> n >> q;
    // get arr
    for(int i=1; i<=n; i++){
        cin >> in;
        arr[i] = in;
    }
```

```
// query
for(int j=0; j<q; j++){
    max = -1;
    cin >> l >> r;
    for(int k=l; k<=r; k++){
        if(arr[k] > max)
            max = arr[k];
    }
    cout << max << '\n';
}
return 0;
}
```

8 points get ! ! !



暴力

$O(NQ)$

$[N < 10^7]$

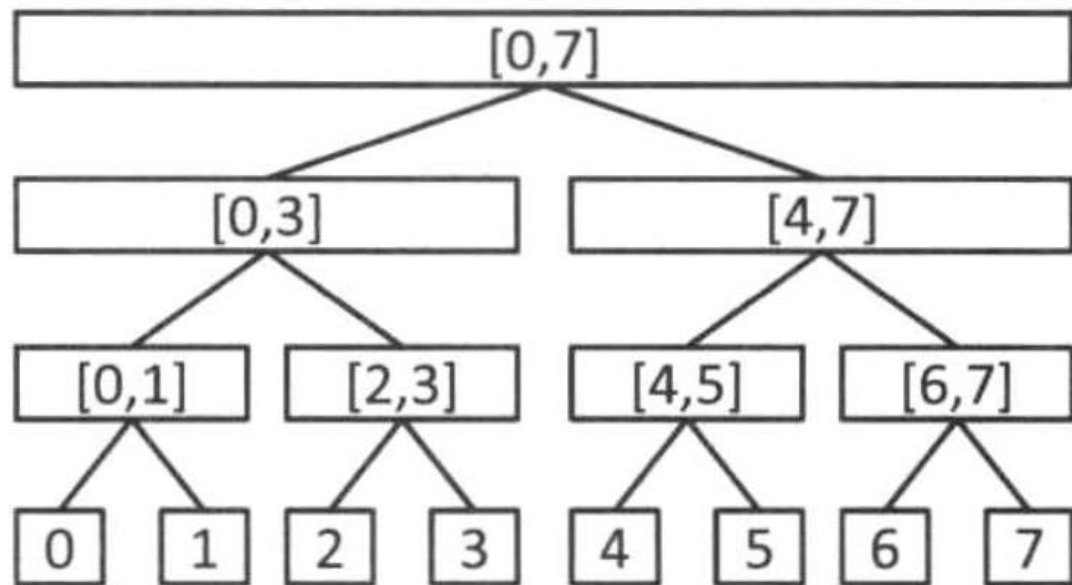
$[Q < 10^6]$

10^{13}

太慢惹吧！ (つдㇿ)

來碗資料結構ㄣ 。。°\(*'∀`)/°。。

D. 普查 正解



線段樹

Google: "演算法筆記 Fake Segment Tree"

D. 普查

```
int main() {
    ios::sync_with_stdio(0); // 加速輸出入
    cin.tie(0); // 加速輸出入
    cin >> n >> m;
    for (int i=1; i<=n; i++)
        cin >> arr[i];

    build(1, n, 1); // 建構線段樹

    int x, y;
    for (int i=0; i<m; i++){
        Max = 0;
        cin >> x >> y;
        if (x>y)
            swap(x, y);
        query(1, n, x, y, 1); // 查詢
        cout << Max << '\n';
    }

    return 0;
}
```

main() 函式

D. 普查

```
int n,m;
long long Max;
long long arr[1000005];    // 10^6
long long seg[1000005*4];  // 10^6 * 4

void build(int l,int r,int pos){

    if (l==r){
        seg[pos] = arr[l];
        return;
    }

    int m = (l+r)>>1;        // (l+r)/2
    int LS = pos<<1;         // 左子樹 index
    int RS = LS|1;           // 右子樹 index

    build(l, m, LS);         // 建構 左子樹
    build(m+1, r, RS);       // 建構 右子樹

    seg[pos] = max(seg[LS], seg[RS]); // 合併左、右子樹
}
```

全域變數

build() 函式

D. 普查

```
void query(int nl, int nr, int tl, int tr, int pos){
    // 查詢左界, 查詢右界, 詢問左界, 詢問右界, 當前seg index

    if ((tl <= nl && nr <= tr)){
        if (seg[pos] > Max)
            Max = seg[pos];
        return;
    }

    int m = (nl+nr)>>1;
    int LS = pos<<1;
    int RS = LS|1;

    if (m < tl) // 詢問左半邊
        query(m+1, nr, tl, tr, RS);
    else if (m >= tr) // 詢問右半邊
        query(nl, m, tl, tr, LS);
    else { // 詢問左、右半邊(跨中點)
        query(nl, m, tl, tr, LS);
        query(m+1, nr, tl, tr, RS);
    }
}
```

query() 函式

線段樹

[build $O(N \log N)$]

[query $O(Q \log N)$]

$O((N + Q) \log N)$

$2.5 * 10^6$

大概390000倍快呢！

(*'∇')~♥

D. 普查

Survey ▶ Scored (48.0 / 100.0)

讚讚 $(\cdot \forall \cdot) \curvearrowright \textcircled{10}$

- 所有經濟狀況指數 $\leq 2^{63} - 1$

開long long是基本常識吧 :3

Survey

▶ Scored (84.0 / 100.0)

崩 (干皿干) 潰

$$l > r$$

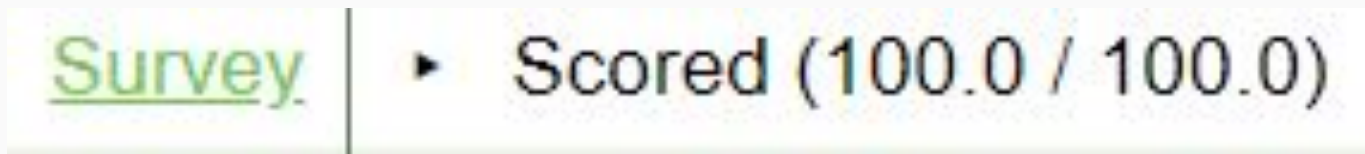
de不到bug～de不到bug～

$$\sigma \overset{\circ}{\nabla} \overset{\circ}{\nabla} \overset{\circ}{\nabla} \overset{\circ}{\nabla} \sigma$$

D. 普查

swap一波 ($\forall$.)

```
if(l > r)
    swap(l, r);
```



100 points get ! (ε:) ^ ^ (.ω.) ^ ^ (:3)

D. 普查

1. 暴力
2. 線段樹, int範圍
3. long long 範圍
4. long long 範圍, 要swap

E. 貧富差距

命題:lys0829

解題狀況

AC: 1人

First Blood: 成俊宏 (02:15:53)

E.貧富差距--題目大意

把一個數列切一切

將每一段總和平方, 在將每一段平方後的結果相加, 求最小

e.x -1 2 4 -3 0 -4

$(-1), (2, 4, -3, 0, -4)$

E.貧富差距

那就都切切看阿!!

每一個間格都可以選擇切或不切, 假設長度是N

這樣有 $2^{(N-1)}$, N最大**10000**

2^{9999} 一秒怎麼可能跑的完

E.貧富差距--子任務1、子任務2

每個數字都 ≥ 0

仔細想一下，你就會發現，當全部都 ≥ 0 時，有一個性質

每個數字**單獨**一定會最好

$$(A+b)^2 = A^2 + 2AB + B^2 \geq A^2 + B^2$$



E.貧富差距--子任務1、子任務2

於是.....

› 子問題 1

(7 / 7)

› 子問題 2

(0 / 10)

記得開long long

GET 17 POINTS

› 子問題 1

(7 / 7)

› 子問題 2

(10 / 10)

› 子問題 3

(0 / 27)

› 子問題 4

(0 / 37)

› 子問題 5

(0 / 19)

E.貧富差距--子任務3

$N \leq 25$

N很小, 看來應該真的可以每個都切切看

```
Long Long calc(int r){
    Long Long ans = p2(add[r]);
    for(int i=r-1;i>=1;i--){
        ans = min(ans,calc(i)+p2(add[r]-add[i]));
    }
    return ans;
}
```



› 子問題 1

(0 / 7)

› 子問題 2

(0 / 10)

› 子問題 3

(27 / 27)

› 子問題 4

(0 / 37)

› 子問題 5

(0 / 19)

Dynamic Programming

沒錯就是DP！！

定義狀態為

$dp[i]$ = 從數列第一項到第 i 項的最小是多少

E.貧富差距

狀態-> $dp[i]$ = 從數列第一項到第 i 項的最小是多少

那轉移呢?

可以將數列從尾巴切一段下來, 切下來的就綁在一起當成一段, 剩下的再丟下去遞迴

只要將每種長度的尾巴都試過, 就能找到最好的那種切法

E.貧富差距

於是乎，你可能就寫出這樣的東西.....

```
inline long long calcadd(int l,int r){
    long long ans=0;
    for(int i=l;i<=r;i++){
        ans+=num[i];
    }
    return ans;
}

long long calc(int r){
    test++;
    if(used[r])return dp[r];
    used[r]=true;
    dp[r] = p2(calcadd(1,r));
    for(int i=r-1;i>=1;i--){
        dp[r] = min(dp[r],calc(i)+p2(calcadd(i+1,r)));
    }
    return dp[r];
}
```


E.貧富差距

什麼!!!這一定是judge壞了.....

```
inline long long calcadd(int l,int r){
    long long ans=0;
    for(int i=l;i<=r;i++){
        ans+=num[i];
    }
    return ans;
}

Long Long calc(int r){
    test++;
    if(used[r])return dp[r];
    used[r]=true;
    dp[r] = p2(calcadd(1,r));
    for(int i=r-1;i>=1;i--){
        dp[r] = min(dp[r],calc(i)+p2(calcadd(i+1,r)));
    }
    return dp[r];
}
```

子問題 1		(7 / 7)
子問題 2		(0 / 10)
子問題 3		(27 / 27)
子問題 4		(37 / 37)
子問題 5		(0 / 19)

#	評測結果	詳細資訊	執行時間	記憶體使用量
31	無法得分	超過時間限制	1.979 秒	768 KiB
32	無法得分	超過時間限制	1.981 秒	640 KiB



E.貧富差距

每次要算第1項到第r項的總和，還要用迴圈一個一個加總實在是太慢了

那要怎麼辦呢??

應該很多人都知道**前綴和**吧，不知道沒關係，你一定學過

高一下數學--數列與**級數**

事先開一個**陣列**存下第1項到第i項的**總和**， $A_i = S_i - S_{i-1}$

E.貧富差距

```
long long calc(int r){
    test++;
    if(used[r])return dp[r];
    used[r]=true;
    dp[r] = p2(add[r]);
    for(int i=r-1;i>=1;i--){
        dp[r] = min(dp[r],calc(i)+p2(add[r]-add[i]));
    }
    return dp[r];
}
```

› 子問題 1

(7 / 7)

› 子問題 2

(10 / 10)

› 子問題 3

(27 / 27)

› 子問題 4

(37 / 37)

› 子問題 5

(19 / 19)

F. 好哈拷貝

命題：潘淙軒

解題狀況

AC: 0人

First Blood: **NULL** (00:00:00)

Highest Score : 33pts by. **周暘典**

題目大意

比較兩字串中字詞的重複次數

子任務2

本子任務的字詞長度為 1,

使用暴力法將字母一個一個比對後,

將相同的字母存入自 26(a-z)個陣列即可

時間複雜度: N^2

10分get

子任務1

stl_map

使用<string,int>即能儲存字串

然後使用暴力法將字串交由 map處理

即能得到另外 15分

時間複雜度: n^3

```
1  #include<bits/stdc++.h>
2  using namespace std;
3  map<string ,int> mp;
```


子任務3

```
hash_a=( (hash_a%mod*mul+mod)%mod + (ull)a[i+j] )%mod;
```

hashing+map

若能想到將字串轉成數字作表示 (abcd->1234)

對於每個字詞比對時只需複雜度 $O(1)$ 即可完成

時間複雜度: $(n^2) * \log n$

子任務4

rolling_hash+map

若能想到hashing可以用

$a \cdot x^n + b \cdot x^{(n-1)} + \dots$ 的多項式表達

則時間複雜度則能降低成 $n \log n$

然後就滿分了

註:詳情請搜尋 "Rabin-Karp rolling hash"

```
for(int i=0;i<k;i++)power=(power*mul%m+m)%m;
ull hash_a=0,hash_b=0;
for(int i=0;i<k;i++)hash_a=((hash_a*mul+a[i])%m+m)%m;
for(int i=0;i+k<=al;i++){
    //cout<<hash_a<<endl;
    if(mp.count(hash_a)>0)mp[hash_a]++;
    else mp[hash_a]=1;
    hash_a=((hash_a*mul%m+m)%m+a[i+k]-(a[i]*power%m+m)%m+m)%m;
}
```

G.最長的電影

命題：許明盛

解題狀況

AC: 0人

First Blood: **NULL** (00:00:00)

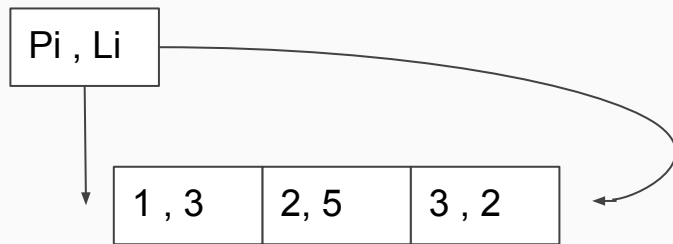
Highest Score : 57pts by. **周暘典**

題目大意

總之題目會依序給你很多數對 (P_i, L_i)

要做的事情很簡單，製造一個數對列，每次新數對只能由列的頭或尾進入或不進入，並保證數對列中的 P_i 是遞增的。

讓數對列的 L_i 加總為最大，並輸出其值



$P_i: 1 < 2 < 3 \dots$

關於子任務四

因為 P_i 是遞增或遞減的, 故全部加起來
就可以了

請注意 `long long int`

YA! 我拿到了6分

暴力至上

既然只有放前、放後、不放，那就每種都試一次吧

複雜度 $O(3^n)$

```

11 T,n,mlis[N],Mlis[N],ar[N],l[N],ans=0;
11 solve(ll x,ll leng,ll f,ll b){
    if(x>=n) return leng;
    ll r1=0,r2=0,r3=0,r;
    if(ar[x+1]>=b) r1=solve(x+1,leng+l[x+1],f,ar[x+1]);
    else if(ar[x+1]<=f) r2=solve(x+1,leng+l[x+1],ar[x+1],b);
    r=max(r1,r2);
    r3=solve(x+1,leng,f,b);
    r=max(r,r3);
    return r;
}

```

共拿到18分 剩餘測資>>>>>>>>>>TLE

解法一

試想第 i 數對為第一個放入數對列中的數對,

那它所能得到的 L_i 總和最大值就來自兩部分:

1.(P_i, L_i) , (P_a, L_a) , (P_b, L_b) , (P_c, L_c).....

$i < a < b < c < \dots$ 且 $P_i < P_a < P_b < P_c < \dots$

2.(P_f, L_f) , (P_e, L_e) , (P_d, L_d) , (P_i, L_i).....

$i < d < e < f < \dots$ 且 $P_i > P_d > P_e > P_f > \dots$

由於 $a, b, c, d, e, f, \dots$ 皆大於 i , 也就是數對

$a, b, c, d, e, f, \dots$ 皆要在數對 i 後輸入

1. 在數對 $a, b, c, d, e, f, \dots$ 中找子數對列其 P 遞增且 L 總和為

最大

2. 在數對 $a, b, c, d, e, f, \dots$ 中找子數對列其 P 遞減且 L 總和為

最大

解法一

找遞增遞減子序列且總和為最大, 有沒有覺得很熟悉?

沒錯! 就是 **LIS!**

只不過要遞增遞減都找, 且因為找的是 P_i 後的數對列, 所以要從數對列尾端找回來

數對 i 的 L 總和最大值 = 數對 i 後的遞增 LIS + 數對 i 後的遞減 LIS + L_i

AC!

```
#include <bits/stdc++.h>
#define N 5001
using namespace std;
typedef long long ll;

ll T,n,mli[N],Mli[N],ar[N],l[N];
int main()
{
    cin>>n;
    for(int i=0;i<n;i++){
        cin>>ar[i]>>l[i];
    }
    ll ans=0;
    for(int i=n-1;i>=0;i--){
        mli[i]=l[i];
        Mli[i]=l[i];
        for(int j=i+1;j<n;j++){
            if(ar[j]>ar[i]){
                Mli[i]=max(Mli[i],Mli[j]+l[i]);
            }
            else if(ar[j]<=ar[i]){
                mli[i]=max(mli[i],mli[j]+l[i]);
            }
        }
        ans=max(ans,Mli[i]+mli[i]-l[i]);
    }
    cout<<ans<<"\n";

    return 0;
}
```

解法二？

我沒想到...

(順帶一提，解法三、四我也沒想)

但畢竟是題目是DP，可能有更多解法，有興趣者可以想想看

H.小黑的芳心

命題：林宇翔

解題狀況

AC: 2人

First Blood: 成俊宏 (02:15:25)

H.小黑的芳心--題目大意

事先決定好一些集合

根據這些集合內是否含有目標數字, 精確猜出目標數字

IOI style題型

採用連續函數評分



H.小黑的芳心--暴力至上

那很簡單阿，就每個洞只放一張卡片

每個洞都放不同的卡片

如此一來，只要知道哪個洞有，就知道答案了

H.小黑的芳心--暴力至上

那很簡單阿，就每個洞只放一張卡片

每個洞都放不同的卡片

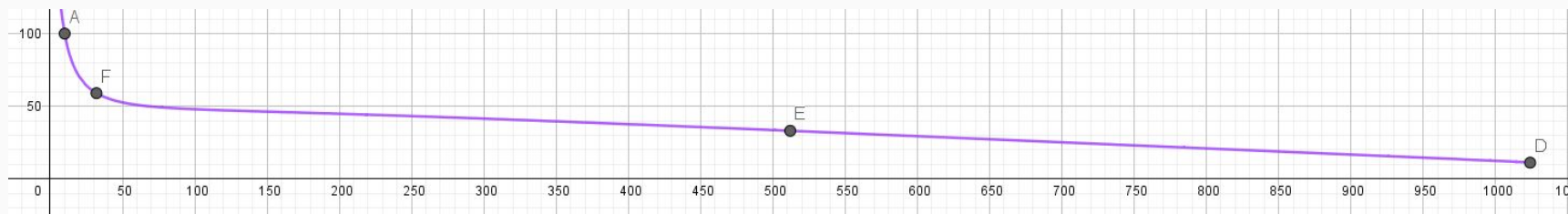
如此一來，只要知道哪個洞有，就知道答案了

子問題 1				(11.15 / 100)	
#	評測結果	詳細資訊	執行時間	記憶體使用量	
1	部分正確		0.014 秒	2.13 MiB	
2	部分正確		0.014 秒	2.13 MiB	
3	部分正確		0.015 秒	2.13 MiB	
4	部分正確		0.021 秒	2.13 MiB	
5	部分正確		0.023 秒	2.13 MiB	
6	部分正確		0.013 秒	2.13 MiB	

好像有點低

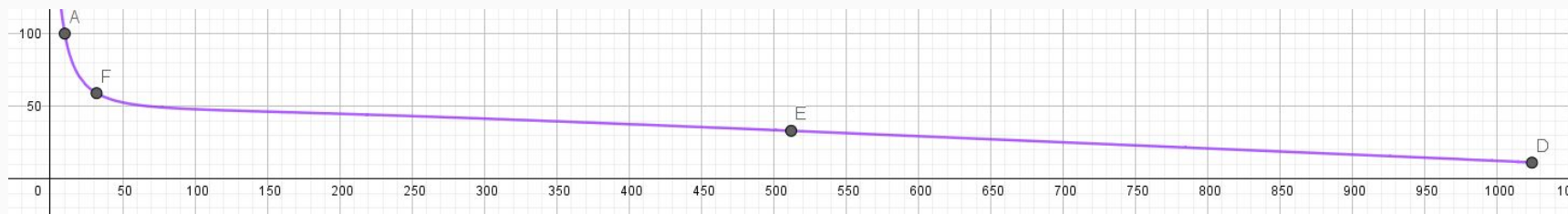
H.小黑的芳心

來看看精心設計的評分函數



H.小黑的芳心

來看看精心設計的評分函數



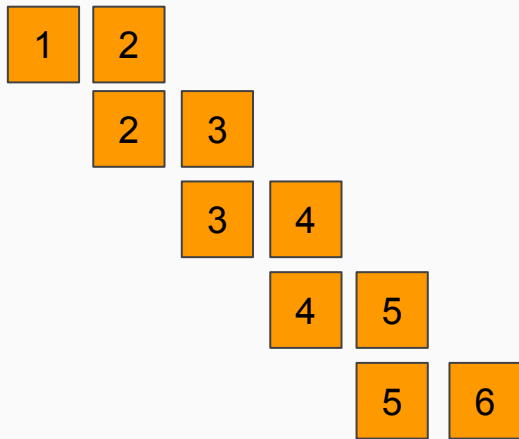
剛剛的做法使用了1024個洞穴

分數大約是11分



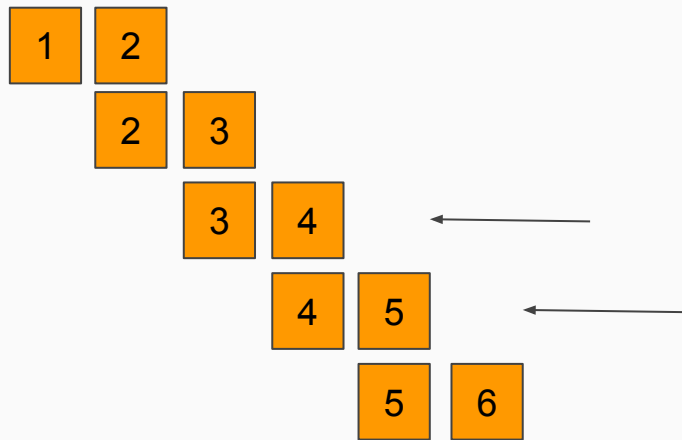
H.小黑的芳心--試著拿更高的分數

如果一個洞穴放兩張卡片呢?



H.小黑的芳心--試著拿更高的分數

如果一個洞穴放兩張卡片呢？



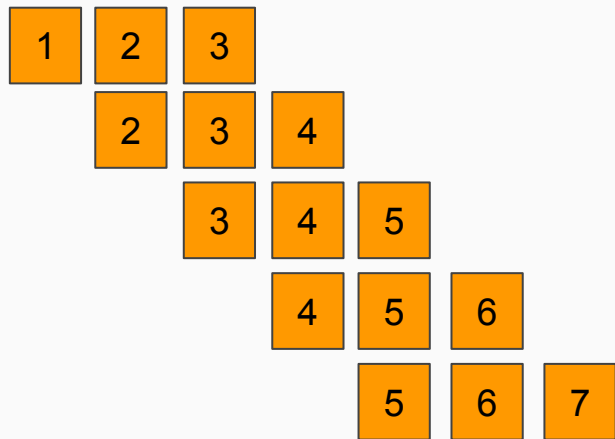
假設答案是4

取交集就找出來了

這樣只要1023個洞

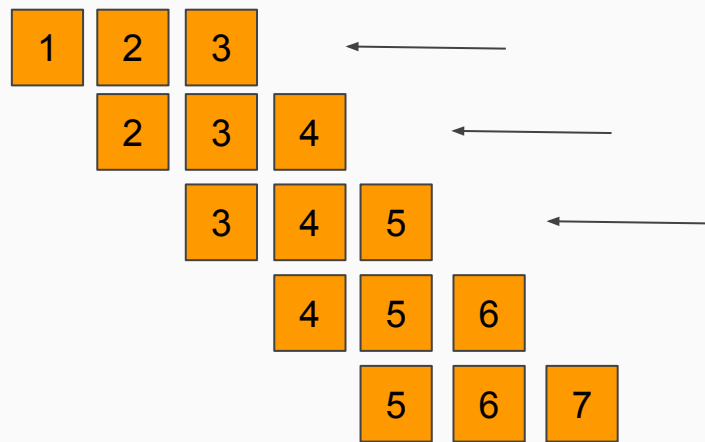
H.小黑的芳心--試著拿更高的分數

那如果一個洞穴變成放三張卡片呢?



H.小黑的芳心--試著拿更高的分數

那如果一個洞穴變成放三張卡片呢？



假設答案是3

也是取交集就找出來了

這樣只要1022個洞

H.小黑的芳心--試著拿更高的分數

如此下去

可以發現, 我們能夠只使用**513**個洞穴就得到答案

▼ 子問題 1				(33.07 / 100)	
#	評測結果	詳細資訊	執行時間	記憶體使用量	
1	部分正確		0.000 秒	128 KiB	
2	部分正確		0.000 秒	128 KiB	
3	部分正確		0.000 秒	128 KiB	

GET 33.07 POINTS



H.小黑的芳心--試著拿更高的分數

好像還是有點低QAQ

再來想想另一個解法吧!

1 2 3 4

5 6 7 8

9 10 11 12

13 14 15 16

1 5 9 13

2 6 10 14

3 7 11 15

4 8 12 16

```
1 #include "guess.h"
2 int main() {
3     #define o0 Addnum
4     #define Oo Ask
5     #define o0Oo SubmitAns
6     Init();
7     for(int i=0;i<32L;i++)for(int j=0;j<32;j++)o0(i,i*32+j);
8     for(int i=0L;i<32;++i)for(int j=0;j<32;++j)o0(32+i,i+j*32);
9     SubmitSet();
10    short one=-1,two=-1;
11    for(int i=0;i<32L;i++)if(Oo(i))one=i;
12    for(int i=0L;i<32;++i)if(Oo(i+32))two=i;
13    o0Oo(one*32+two);
14    return 0;
15 }
16
```

H.小黑的芳心--試著拿更高的分數

可以發現，我們能夠只使用64個洞穴就得到答案

▼ 子問題 1

(50.45 / 100)

#	評測結果	詳細資訊	執行時間	記憶體使用量
1	部分正確	GET 50.45 POINTS	0.000 秒	2.13 MiB
2	部分正確		0.000 秒	2.13 MiB
3	部分正確		0.000 秒	2.13 MiB



H.小黑的芳心--試著拿更高的分數

可惡，都搞成這樣了還只有**50**分

偷偷告訴你

想拿**100**分只能用**10**個洞



H.小黑的芳心--試著拿更高的分數

10跟1024是不是有什麼關係阿

$$2^{10} = 1024$$

H.小黑的芳心--試著拿更高的分數

10跟1024是不是有什麼關係阿

$$2^{10} = 1024$$

你又猜對了, 是二分搜

H.小黑的芳心--試著拿更高的分數

使用二進位, 算出每個位元是0或1

每個洞就放二進位某一位是1的所有數字

H.小黑的芳心--試著拿更高的分數

使用二進位，算出每個位元是0或1

每個洞就放二進位某一位是1的所有數字



子問題 1

(100 / 100)

#	評測結果	詳細資訊	執行時間	記憶體使用量
1	全部正確		0.000 秒	2.13 MiB
2	全部正確		0.004 秒	2.13 MiB
3	全部正確		0.000 秒	2.13 MiB
4	全部正確		0.000 秒	2.13 MiB
5	全部正確		0.000 秒	2.13 MiB

恭喜你拿到100分

挑戰371分吧!!!