

# 建中校隊初選題解

排版陽春請見諒

hansonyu123

# 入門

- 用int/unsigned int/long/unsigned long/long long儲存a跟b，直接比大小
  - 51分
- 用unsigned long long 儲存a跟b，直接比大小
  - 87分
- 用string儲存a跟b。如果長度不一樣，比較長度；如果長度一樣，比較字典序
  - 100分

# 肺噬數列

```
1  #include<bits/stdc++.h>
2
3  using namespace std;
4
5  int fibo(int s)
6  {
7      if(s == 1) return 1;
8      if(s == 2) return 2;
9      return fibo(s-1)+fibo(s-2);
10 }
11
12 int main()
13 {
14     int t;
15     cin>>t;
16     int now = 1;
17     for(int i = 1; i <= t; i++)
18     {
19         for(int j = 0; j < fibo(i); j++)
20         {
21             cout<<now;
22             now = (now+1)%3;
23         }
24         cout<<endl;
25     }
26 }
```

TLE 60分

# 肺噬數列

```
1  #include<bits/stdc++.h>
2
3  using namespace std;
4
5  int fibo(int s)
6  {
7      if(s == 1) return 1;
8      if(s == 2) return 2;
9      return fibo(s-1)+fibo(s-2);
10 }
11
12 int main()
13 {
14     int t;
15     cin>>t;
16     int now = 1;
17     for(int i = 1; i <= t; i++)
18     {
19         int temp = fibo(i);
20         for(int j = 0; j < temp; j++)
21         {
22             cout<<now;
23             now = (now+1)%3;
24         }
25         cout<<endl;
26     }
27 }
```

AC!

# 肺噬數列

```
1  #include<bits/stdc++.h>
2
3  using namespace std;
4
5  int fibo[31];
6
7  int main()
8  {
9      int t;
10     cin>>t;
11     int now = 1;
12     fibo[2] = 2, fibo[1] = 1;
13     for(int i = 1; i <= t; i++)
14     {
15         if(i >= 3) fibo[i] = fibo[i-1]+fibo[i-2];
16         for(int j = 0; j < fibo[i]; j++)
17         {
18             cout<<now;
19             now = (now+1)%3;
20         }
21         cout<<endl;
22     }
23 }
```

# 握手宴會 Subtask 1

- 直接開一個 $n$ 乘 $n$ 的陣列，記錄兩個人之間有沒有握手
  - 空間複雜度 $O(n^2)$ ，時間複雜度 $O(m+q)$
  - MLE 11分

# 握手宴會 Subtask 2

- 用vector將圖存起來。之後每次詢問i跟j有沒有握過手時，都直接到edge[i]找有沒有j
  - 空間複雜度 $O(m+n)$ ，時間複雜度 $O(mq)$
  - TLE\MLE 13分

# 握手宴會 Subtask 3

- 邊數很少的時候，需要考慮的點只有 $2m$ 個，所以可以直接開 $2m$ 乘 $2m$ 的鄰接矩陣
- 空間複雜度 $O(m^2)$ ，時間複雜度 $O(m+q)$ 
  - MLE 24分



# 握手宴會 Subtask 4

- 直接把每條邊當成一個pair，先放進vector後排序。每次詢問時直接二分搜就好
- 空間複雜度 $O(m)$ ，時間複雜度 $O((m+q)\lg m)$   
– 100分

# 填方格遊戲

- **KEY POINT:** 不管以什麼順序填數字，結果都不會變
- 所以不管怎麼玩，會輸的就是會輸
- 乾脆把整個棋盤數字一次填好再找遞增數列
- 遞增數列找法：紀錄某個位置最多往前幾項還是遞增數列
- 複雜度 $O(M^2)$

**MLE 91分**

# 填方格遊戲 Subtask 5

- 空間存不下整個方格
- 每次只需要存一行的結果，以計算下一行的值，便可將之前存的結果丟棄，留下需要的東西
- 空間複雜度降為 $O(M)$

**AC!**

# 填方格遊戲 Subtask 5

- $D[a][b] = D[0][0] \wedge D[0][b] \wedge D[a][0]$
- 這樣就不用滾動了

**AC!**

# 俄羅斯娃娃

- 先將二維問題轉為一維問題
- 對其中一維排序
  - 看起來很像最長遞增子序列(LIS)問題
  - 但 $w_i$ 相等時不能選
  - 解決方法： $w$ 由小排到大， $w$ 一樣的話 $h$ 由大排到小
  - 和LIS等價，複雜度 $O(n \lg n)$
  - 100分

# 尋找蘿莉 Subtask 1

- 二分搜是最佳策略
- 可以直接計算出答案， $O(1)$
- 也可以直接模擬， $O(\lg n)$
- 對於其它subtask來說，二分搜不一定會最好
  - 9分

# 尋找蘿莉

- 每次都將蘿莉分成兩堆S1和S2，假設SS級蘿莉在S1的機率是W1，在S2的機率是W2
- 如果已知SS級蘿莉在S1中時最佳期望值是A，在S2中時最佳期望值是B，那麼答案就是 $1+P1A+P2B$
- 兩邊同乘 $\Sigma P$ ，用X1、X2、X來表示：
- $X=\Sigma P+X1+X2$
- 要找到S1和S2使得X最小

## 尋找蘿莉 Subtask 2

- 對於每個S，都暴搜所有的S1和S2，繼續遞迴下去
- 時間複雜度為 $O(n!)$ ，空間複雜度 $O(n)$ 
  - 13分



# 尋找蘿莉 Subtask 3

- 把每個可能的S的答案存好，以後就不用再算一次了
- 複雜度為 $O(3^n)$ ，空間複雜度 $O(2^n)$ 
  - 28分

# 尋找蘿莉 Subtask 4

- 將策略圖畫出。以範測為例

