建中校隊複選題解

排版陽春請見諒

這次我每個（重要的）

subtask都有寫（有截圖證明）

所以不要以為分數拿不到

一個數列

- 不要CE
  - 1分
  - 應該不會有人沒拿到這1分吧
- Brutal Force
  - 複雜度O(Σ(b-a)+Q)
  - 30分，這30分簡直送分

43252	1227	玩泥巴:3	19785	16816	TLE	c++11	358 Bytes	30

一個數列

- 先將奇數跟偶數分開看兩個數列
- 對於兩個數列，需要做區間加值以及單點詢問兩種操作
  - 然而詢問一定在加值後出現
- 維護a[i+1]-a[i]序列，預處理O(N)，每次加值只需要O(1)，而且變換回原數列也只需要O(N)，詢問只需要O(1)。總複雜度O(N+M+Q)
  - AC。只要不要寫BIT/線段樹基本上這題送分

43275	1227	玩泥巴:3	3428	16736	AC	c++11	581 Bytes	100

- 線段樹/BIT (?
- QQ測資沒生好 寫BIT的這次先放過(??

8		ckcon15_22	ckcon15_22	57	0	0	0	0	0	57
9		ckcon15_04	ckcon15_04	57	0	0	0	0	0	57

尋找蘿莉第二彈

- Subtask 1初選就出現過了，13分不拿就算了
- 初選講過每次要把蘿莉分成兩堆使得兩堆答案之和最小，答案就是這個最小值再加上機率總和。這次只能把區間拆成兩個區間，直接DFS複雜度是$O(C_{n-1})$，其中C是卡特蘭數。C_{17}大約是1.4*10^8。
  - TLE/MLE 34分。這34分真的佛心來著。

43294	1639	玩泥巴:3	21881	1344	MLE	c++11	513 Bytes	34

尋找蘿莉第二彈

- 要拿到100分也很簡單，就是把答案存起來(DP)。狀態數O(n^2)，轉移O(n)，共O(n^3)
- 這100分是送分的

43295	1639	玩泥巴:3	6901	6012	MLE	c++11	556 Bytes	100

- 不過剩下的50分需要DP優化的技巧：四邊形優化。我們在第五次培訓會講到。
- 為了證明還是能AC(複雜度O(n^2))：

43311	1639	玩泥巴:3	1280	126136	AC	c++11	900 Bytes	150

喵喵的旅程

- 你會發現如果知道「鄉鎮最遠距離」最大和最小的兩個城市是誰的話，直接計算直徑大小就好。因為這個教過(兩次DFS)，我就不處理前面的subtask了。
- 重點是如何找到最大和最小。顯然可以n-1次找最大，n-1次找最小，共2(n-1)次。這樣：

43325	1164	玩泥巴:3	9038	75812	WA	c++11	1.06 KB	53

- 這53分沒拿到的話以後上課專心一點

喵喵的旅程

- 前面用了約2n次。事實上約1.5n次就夠了
- 先兩兩捉對撕殺。如果比較小的就不可能是最大值，比較大的就不可能是最小值。比較約0.5n次
- 把有潛力是最大/最小值的挑出來再分別求最大/最小值。分別比較約0.5n次。共約1.5n次。這是最好的了。這樣就AC囉。

43338	1164	玩泥巴:3	8895	78208	AC	c++11	1.4 KB	100

- 之前筆試講過這東西，所以這整題都是來送分的。沒AC的以後上課要專心。

```cpp
int main()
{
    int n=init();
    for(int i=0; i<n; i++) city[i]=i;
    int most, fewest;
    if(query(0,1)==1)
    {
        most=0;
        fewest=1;
    }
    else
    {
        most=1;
        fewest=0;
    }
    for(int i=2; i<n; i++)
    {
        if(query(i,most)==1)
        {
            most=i;
            continue;
        }
        else
        {
            if(query(i,fewest)==0)
            {
                fewest=i;
            }
        }
    }
```

```cpp
int main()
{
    int n=init();
    for(int i=0; i<n; i++) city[i]=i;
    int most, fewest;
    auto tmp = minmax_element(city, city+n, query);
    most = *(tmp.first), fewest = *(tmp.second);
```

Bonus time (Subtask 1)

- 你會發現如果兩個字串中的a都不超過k個，那麼答案就是2*min(a_1, a_2)，這裡a_1和a_2是兩個字串的a的個數。而如果兩個字串的a的個數都有k+3個，就不用煩惱了
- 問題就只發生在min(a_1, a_2)=k+1以及 min(a_1, a_2)=k+2的極少數情況(m跟n要夠小但min(m,n)>min(a_1, a_2)，滿稀有的)
- 好險測資中沒有這兩個情況(?)所以就不小心拿到13分了(真唬爛)

43383	1427	玩泥巴:3	38	1268	WA	c++11	350 Bytes	13

Bonus time

- k很小，說不定可以用dp
- 狀態存法：dp[i][j][k]，代表第一個字串前i個，第二個字串前j個字元，可以寫至多k個子字串的答案是多少
- 在轉移時發現還需要紀錄結尾有沒有同時被寫到(才能知道k要不要加1)，所以再加一維[0 or 1]代表
- 轉移式很好寫，略。複雜度O(kn^2)

43384	1427	玩泥巴:3	857	79456	MLE	c++11	1004 Bytes	87

Bonus time

- BTW，如何脫離87
- int->short
- MLE 90

Bonus time

- dp陣列MLE了！
- 仔細觀察可以發現要計算dp[i][j][k][0 or 1]的答案，只需要dp[i+1][j][k-1 or k][0 or 1], dp[i][j][k-1 or k][0 or 1]以及dp[i][j+1][k-1 or k][0 or 1]，所以可以滾動DP。空間複雜度降為O(kn)。

43556	1427	玩泥巴:3	7093	2048	AC	c++11	1.23 KB	100

Phh多層次傳銷公司 (Subtask 1)

- 轉換問題：動態詢問帶權有根樹的一個子樹的邊權和(記得乘兩倍再輸出)
- Subtask 1中邊權不會改變，只要算一次所有人的子樹邊權和就好
- 算法：s的答案=Σ(s的小孩的答案+邊權)
  - 樹分治(樹DP)，第五次上課會仔細講
- 時間複雜度O(N+Q)

43588	1228	玩泥巴:3	17061	86092	WA	c++11	968 Bytes	7

Phh多層次傳銷公司 (Subtask 2)

- 套用剛剛的想法，如果算完之後邊權有更改過…那就重算一次就好啦！
- 時間複雜度O(NQ)

43587	1228	玩泥巴:3	47808	86080	TLE	c++11	952 Bytes	24

- 另一種解法：每次更新一條邊的時候，只有他的祖先們的答案會受影響
- 把更改紀錄存好。每次詢問時看一下有哪些更改紀錄會影響到他(快速查詢祖先關係)
- 時間複雜度同樣是O(NQ)

Phh多層次傳銷公司 (Subtask 3)

- 結合Subtask 2中的兩個想法
  - 如果更改紀錄不多的話，直接看看更改紀錄的每個人對答案影響多大。複雜度O(k)(k是更改紀錄個數上限)
  - 如果更改紀錄超過k的話，直接砍掉重練。複雜度O(N)
- 總複雜度O(Qk+N(Q/k))，在k是O(√N)時複雜度最好(算幾不等式)，複雜度O(Q√N)
  - 此法稱為「平方分割」
- k取100的時候：

43605	1228	玩泥巴:3	32791	91124	MLE	c++11	1.32 KB	46

Phh多層次傳銷公司 (Subtask 4)

- 平方分割的常數很大，不過如果k取得適當的話可以有效地減少常數
- 經過試驗之後，發現k=600的時候可以把Subtask 4也解決掉←懶得試最好在什麼時候

43604	1228	玩泥巴:3	26504	91124	MLE	c++11	1.32 KB	71

- 這種解法拿71分算是佛心來著了@@

Phh多層次傳銷公司

- 之前在處理「快速判斷是否是祖先」的時候，要用DFS的遍歷順序作為判斷依據
- 小小觀察：一棵子樹的所有點的編號剛好是一段區間。所以求答案等同是求一段區間內的數字和，而要支援可以更改單一一個數字的操作
- 使用BIT(第六次會講)，複雜度O(N+QlgN)
  - 此技巧稱為「樹壓平」

43624	1228	玩泥巴:3	17521	83764	AC	c++11	1.23 KB	100

賣萌大賽(Subtask 1)

- 直接暴搜…沒什麼好說的
  - 時間複雜度O(n*2^m)
  - 這11分很送分

43735	1830	玩泥巴:3	117	1236	MLE	c++11	770 Bytes	11

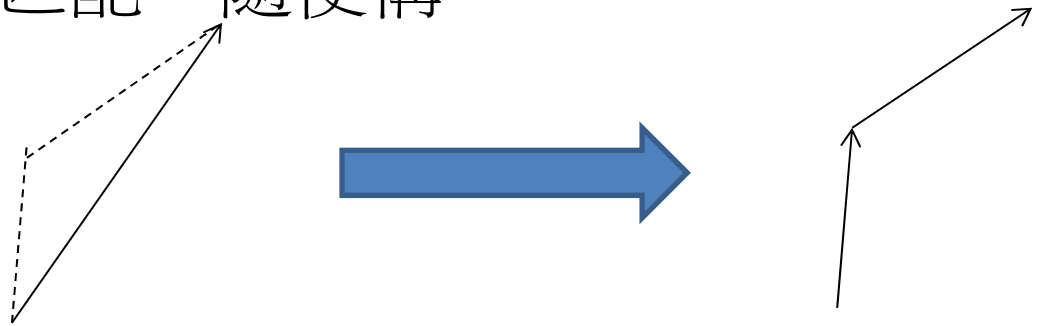
賣萌大賽(Subtask 2)

- 轉換問題：有一個無向圖，有一些邊的邊權是1，其它是2。每個點的邊權和都是奇數。要把這個無向圖賦向使得每個點的出度和入度之差恰為1。
- n很小
- 如果兩個點之間有同樣權重的邊，直接令他們是反向消去即可。餘下至多n*(n-1)條
- 複雜度$O(n*2^{n(n-1)})$

43758	1830	玩泥巴:3	31712	2312	TLE	c++11	1.47 KB	18

賣萌大賽(Subtask 3)

- 沿用剛剛的想法，試圖把邊變少
- 仔細觀察可以發現如果有一個點的degree超過2，可以把兩條邊黏起 來變一條邊(且保持各點邊權和是奇數，所以仍然有解)，再繼續求解即可。最後變完美匹配，隨便構



- Submit後發現連Subtask 4一起過了，這並不是巧合，可以想想看為什麼

43796	1830	玩泥巴:3	18360	36068	WA	c++11	1.46 KB	42

賣萌大賽(Subtask 4)(另解)

- 可以發現Subtask 4中的邊一定是1 2交錯的鏈或者環
- 直接構出賦向的方法

43811	1830	玩泥巴:3	18381	119796	MLE	c++11	1.88 KB	28

賣萌大賽

- 整題的解法也沒什麼。就先用Subtask 3的做法把共用端點權重相同的邊縮起來，就變成Subtask 4的情況了。
  - 小subtask激發靈感

42426	1830	玩泥巴:3	18790	49644	AC	c++11	3.01 KB	100

Submitter:	ckcon15_14	Compiler:	c++11	Code Length:	4 Bytes

```
1 | jizz
```

送分統計(?)

- pA:S1-S2 30分
- pB:S1-S3 100分
- pC:全部 100分
- pD:0分
- pE:0分
- pF:S1 11分
- 簡單來說沒拿到241分的可以去面壁了
  - 因為太慘了所以只好叫你們面白板…不然沒人聽我講話@@

Code存放地(C++11)

- pA
  - S1-S3: http://codepad.org/hiTltnYM
  - AC: http://codepad.org/aKlXKddS
  - AC: http://codepad.org/xqGI6557
- pB
  - S2: http://codepad.org/PAPXY0P4
  - S1-S3: http://codepad.org/jI57VDeY
  - AC: http://codepad.org/lAmCSCFt
  - AC: http://codepad.org/miUwLgsW

Code存放地(C++11)

- pC
  - S1-S4: http://codepad.org/hnZ2NUTG
  - AC: http://codepad.org/AeS4I9Lx
  - AC: http://codepad.org/y5yP2m6R
- pD
  - S1: http://codepad.org/JmotQ2zU
  - S1-S4: http://codepad.org/k3XsE91o
  - AC: http://codepad.org/cnoRY8Y5
  - AC: http://codepad.org/zg3X6qIU

Code存放地(C++11)

- pE
  - S1: http://codepad.org/fpcTuOgm
  - S1-S2: http://codepad.org/MKfNTXh0
  - S1-S3: http://codepad.org/RuGtxBZf
  - S1-S4: http://codepad.org/kbo23Ft9
  - AC: http://codepad.org/wQjyrl0s
  - AC: http://codepad.org/IhtXlJ5K

Code存放地(C++11)

- pF
  - S1: http://codepad.org/KE2WE7qa
  - S1-S2: http://codepad.org/qp8Fdmyw
  - S3-S4: http://codepad.org/x9C3URNr
  - S4: http://codepad.org/9vIyTLTt
  - AC: http://codepad.org/R1ZnQ9FE
  - AC: http://codepad.org/QjugCi0m