建中校內第四次模擬賽題解

Yihda Yol, hansonyu123

pA. 化學磁牛 Subtask 1, 2, 3, 4

- Subtask 1：隨便亂找答案即可，複雜度估計略。
- Subtask 2：將第i個位置吸回來的機率令為Q_i，得到關係式
$$Q_i = pQ_{i+1} + (1-p)\,Q_{i-1}$$
- 因此不能吸回來的機率P_i關係式為
$$P_i = pP_{i-1} + (1-p)\,P_{i+1}$$
  - 想像成「可以吸回來」的版本倒過來看
- 聯立方程可解，複雜度$O(N^3)$。
- Subtask 3,4：由於每個都只和前後兩項相關，可$O(N)$解之。

pA. 化學磁牛

- 不妨假設機率$p < \frac{1}{2}$，可以觀察到相鄰兩項的差要形成一個公比為$\frac{1-p}{p}$的等比數列，才能滿足上頁的關係式
- 以邊界條件代入，可得所求

$$\frac{1-\left(\frac{p}{1-p}\right)^K}{1-\left(\frac{p}{1-p}\right)^N} = \frac{1-\left(\frac{A}{B-A}\right)^K}{1-\left(\frac{A}{B-A}\right)^N} = \frac{(B-A)^{N-K}\left((B-A)^K - A^K\right)}{(B-A)^N - A^N}$$

pA. 化學磁牛

- 仔細分析之後會發現這個式子套在$p > \frac{1}{2}$也是對的
- 但是它不是最簡分數
- $\gcd(a^n - b^n, a^m - b^m) = a^{\gcd(n,m)} - b^{\gcd(n,m)}$
- 用模逆元和快速冪，複雜度$O(\log N)$，AC
  - 記得特判p=1/2的情況，還有這題的mod是$10^8 + 7$
- 本題是建中校隊培訓題，可惜官方解是$O(N)$
  - 數學

pB. Weep Subtask 1, 2

- Subtask 1：暴力枚舉， $O(N^2)$，TLE 8
  - 這個subtask沒拿到一定是你的問題
- Subtask 2：對於每個i，他對答案的貢獻是$\frac{N}{i} - 1$
  - $O(N)$，TLE 30

pB. Weep

- 線性掃過去太浪費時間了
- 其實只要知道對於每個k，有幾個i會滿足$\frac{N}{i} = k$
- 由小到大枚舉$\frac{N}{i}$的可能值，個數即是$i_{now} - i_{last}$
- $O(\sqrt{N})$，AC

- 如果是用枚舉1~$\lfloor\sqrt{N}\rfloor$做的話，要小心$\lfloor\sqrt{N}\rfloor$被重複枚舉到的條件是$\left\lfloor\frac{N}{\lfloor\sqrt{N}\rfloor}\right\rfloor = \lfloor\sqrt{N}\rfloor$，記得扣掉

pC. 普物之氣體分子碰撞 Subtask 1, 2

- 反正不管怎麼樣先暴力解嘛，$O(N^2)$，TLE 35
- WA 7？
- $3 \times (2 \times 10^9)^2 = 1.2 \times 10^{19}$，剛好爆long long，記得unsigned

```
#define INF 10000000000000
    unsigned long long d=INF;
    printf("%llu\n",d);
```

pC. 普物之氣體分子碰撞

- 裸最近點對三維版
  - 二維版本大家都會了（Divide and Conquer或掃描線）
- 三維不會就想辦法轉換成二維
- Divide and Conquer、投影！

pC. 普物之氣體分子碰撞

- 仔細想合併的步驟
- 假設點從平行yz的平面切成兩部分，兩部分的答案已知
- 若答案較小的為d，只需要考慮位於切面距離$< d$的點
- 將這些點投影到yz平面上
  - 原本距離小於d的點對投影後距離仍然會小於d
- 用二維的版本找出所有距離不超過d的點對，再回到三維驗證
  - 不難改，複雜度會是$O((N+M)\log N)$，M是符合條件的點對數

pC. 普物之氣體分子碰撞

- 由於這些點具有一定的稀疏度，合理相信距離不超過d的點對不會太多
- 事實上是$O(N)$，所以原演算法仍然可以在$O(N\log N)$ 時間結束
- 複雜度： $T(N) = 2T\left(\frac{N}{2}\right) + O(N\log N)$
- $T(N) = O(N\log^2 N)$，AC
  - 問題很小的時候可以枚舉以加速

pD. 巨大史萊姆 Subtask 1, 2

- Subtask 1：暴搜，$O(N \times N!)$
- Subtask 2：位元DP，$O(N2^N)$或$O(2^N)$
  - 要壓到$O(2^N)$的話需要位元運算小技巧

- 剩下的兩個subtask要分別用兩個作法，還要合併

pD. 巨大史萊姆 Subtask 3

- 感覺上還是要DP，但是像前面一樣$O(2^N)$太無謀了
- 注意到我們注意的重點只有「現在有幾坨史萊姆」以及「進去了幾隻史萊姆」，以此當狀態的話狀態數$O(N^2)$，貌似可行
  - 要把這種存法的狀態轉回原本的狀態，只要乘上「插入空格」的方法數即可（高中組合）
  - 不過小心K=N的時候需要特判
- 每加入一個史萊姆，可以把兩坨合併，加在一坨的左邊，加在一坨的右邊，或獨立一坨

pD. 巨大史萊姆 Subtask 3

- 轉移式:
  $a(i,j) = (i-1)\,a(i-1,j-1) + 2i\,a(i,j) + (i+1)a(i+1,j-1)$
  - 有$i-1$個空格可以插+有i坨的左邊跟右邊可以選+有$i+1$個空格可以合併
- 轉移複雜度$O(1)$，故總複雜度$O(N^2)$
- 第二維可以滾動，但是暫時沒這個必要
  - MLE 53

pD. 巨大史萊姆 Subtask 4

- 關鍵:
  - 第二維可以滾動
  - 轉移式相當於乘上一個矩陣（齊次線性遞迴）
  - G（矩陣邊長）超小
- 矩陣乘法快速冪即可
  - 不用寫Strassen（寫了可能更慢(?)）
  - 複雜度$O(G^3 \log N)$, MLE 63
- 跟前面一種作法合起來就AC了

pE. 三國殺 Subtask 1

- 直接算！！！

pE. 三國殺 Subtask 2

- 這題看起來DP-able(?)，所以先想想直接DP會發生什麼事
- 假設dp(n, m)代表有n個人，第m個人存活的機率
- 重點是看「第一次砍到第i個人的機率」是多少，才能將問題規模減小
- 第一次砍到第$i+1$個人的機率是第一次砍到第i個人的一半，所以第一次砍到第i個人的機率是$\frac{1}{\frac{1}{2}+\frac{1}{4}+\cdots+\frac{1}{2^N}} \times \frac{1}{2^i} = \frac{1}{1-\frac{1}{2^N}} \times \frac{1}{2^i}$，先簡記為a(n, i)好了。

pE. 三國殺 Subtask 2

- 接著就簡單了
- $dp(n,m) =$

$$\sum_{i=1}^{m-1} a(n,i)dp(n-1,m-i) + \sum_{i=m+1}^{n} a(n,i)dp(n-1,n+m-i)$$

- 狀態數$O(N^2)$，轉移複雜度$O(N)$，總複雜度$O(N^3)$
- TLE/MLE 34

pE. 三國殺 Subtask 3

- 剛剛的狀態數太多了。能不能只靠dp(1, 1)… dp(n, n)計算出dp(n, i)呢?
- 當然可以。如果i≠n，就先看i第一次被閃電判定之後共有k個人被砍了的機率是多少
  - 注意到第i個人被砍了就不用算了

$$dp(n,i) = \frac{1}{2}\sum_{j=0}^{i-1}\binom{i-1}{j}\frac{dp(n-j,n-j)}{2^{i-1}}$$

pE. 三國殺 Subtask 3

- 所以只要計算dp(k, k)就好
- 上面那條式子n跟i都代k的話，對每個dp(k, k)都只要$O(k)$的時間。故總複雜度$O(N^2)$
  - 小技巧：$\binom{i-1}{j}\frac{1}{2^{i-1}}$可用巴斯卡恆等式DP求之以避免精度不足
- TLE 67

pE. 三國殺

- 觀察：$\binom{i-1}{j}\frac{1}{2^{i-1}}$在$j$太小和$j$太大的時候值會非常小
  - 在用巴斯卡恆等式計算$\binom{i-1}{j}\frac{1}{2^{i-1}}$時，如果值小過一個限度eps，就略去不算以減少時間複雜度。
- eps取10^{-22}可AC

pE. 三國殺 另解

- 這個方法只能求出近似解，但可以無限逼近
  - 這裡只做N=M的情況，其它情況是類似的
- 第一輪判定完第N-1號人就留下第N個人的機率：$\left(\frac{1}{2}\right)^{N-1}$
- 第二輪：$\frac{1}{2}\left(\left(\frac{3}{4}\right)^{N-1}-\left(\frac{1}{2}\right)^{N-1}\right)$
  - 第N個人要撐過一次判定(1/2)，前N-1個人在兩次判定至少要死一次(3/4)，但不能第一次就死光(1/2)

pE. 三國殺 另解

- 第k輪： $\frac{1}{2^{k-1}}\left(\left(\frac{2^k-1}{2^k}\right)^{N-1}-\left(\frac{2^{k-1}-1}{2^{k-1}}\right)^{N-1}\right)$
- 可以發現第t輪結束的機率$<\frac{1}{2^t}$，故算到t = 50就很足夠了。N≠M時作法類似，如下式。複雜度O(精度)。

$$\frac{1}{2^{k-1}}\left(\frac{2^k-1}{2^k}\right)^{M-1}\left(\frac{2^{k-1}-1}{2^{k-1}}\right)^{N-M}$$

pF. 喵喵的旅程—續 Subtask 1, 3

- Subtask 1：暴力，複雜度略
- Subtask 3：最小生成樹+LCA直接做，$O((N+M+Q)\log N)$

pF. 喵喵的旅程—續 Subtask 4

- 「A到B的所有可能路徑的所有道路」
- 如果A, B在同一個邊雙連通分量，那麼這個邊雙連通分量的任何一條邊都會被某些路徑經過
- 存在歐拉迴路代表原圖是邊雙連通圖
- 照樣MST，另外記錄修改即可，$O((N+M+Q)\log N)$

pF. 喵喵的旅程—續 Subtask 2

- 如果A, B不在同一個邊雙連通分量
- 把邊雙連通分量縮成點，變成在樹上走簡單路徑
- 對每個邊雙連通分量做MST、記錄修改
- 查詢和修改直接在縮點後的樹上DFS
- $O((QN + M) \log N)$

- 縮成點後的樹，每條邊都要知道它連接的點是邊雙連通分量裡的哪一個，才能好好在MST上查詢

pF. 喵喵的旅程—續 Subtask 5

- 修改一條路徑和查詢一條路徑
- 樹鍊剖分
- $O(N\log N + Q\log^2 N)$

pF. 喵喵的旅程—續

- Subtask 2和5合起來就可以了
  - 說起來好像很簡單(?
- 樹鍊剖分—維護邊雙連通分量和橋的修改（線段樹）
  - 預處理好鍊裡面邊雙連通分量的值，分清楚邊雙連通分量和橋
- 邊雙連通分量—維護MST
- 樹鍊剖分換鍊的時候要查詢接「點」的所屬的MST
  - 所以還要維護最祖先的邊連向邊雙連通分量裡的哪一點
- 查詢的時候好好判case

pF. 喵喵的旅程—續

- $O((N+M)\log N + Q\log^2 N)$，AC
- Coding複雜度$O(crazy)$ (??

- 350行