CODE THANKS FESTIVAL 2018

2018/11/25

satashun

A – Two Problems 解説

- 概要
- A分で解けるB点の問題と、C分で解けるD点の問題がある
- 2問目の方が点が高い (B<=D)
- T分で最大何点取れますか

A – Two Problems 解説

- 2問解ける場合は両方解く
- それ以外は、高々1問しか解けないので、解けるなら2問目を解いた方がいい
- 2問目が解けないなら1問目を解く
- それでも無理なら0点

A – Two Problems 解説

- まとめ
- if A+C<=T : B+D点
- else if C<=T : D点
- else if A <= T : B点
- else : 0点

B – Colored Balls 解説

- 概要
- 赤い玉がX個、青い玉がY個ある (X,Y<=10^9)
- (赤, 青)を(1個, 3個)か(3個, 1個)ずつ減らしていく
- 両方ちょうど0にできるか

B – Colored Balls 解説

- まず、どちらにしても合計は4個ずつ減る
- X + Y が4の倍数ではない時、明らかにNo
- そうでない時、操作は A = (X + Y) / 4 回行われる
- (1個,3個)か(3個,1個)取る操作は、(1,1)を毎回取った上で(0,2)か(2,0)個取る操作
- X – A と Y – A が共に0以上の偶数の場合のみYes

B – Colored Balls 解説

- 別解
- (1個,3個)取る操作をP回、(3個,1個)取る操作をQ回とする
- P + 3Q = X, 3P + Q = Y を解いてもいいです

C – Pair Distance 解説

- 概要
- N人の人がそれぞれ a_i に住んでいます
- 全ての i < j についての |a_i – a_j| の和を求めてください

C – Pair Distance 解説

- まず、愚直に計算すると O(N^2)
- 絶対値が扱い辛い
- a_i をソートすると、i < j について a_j – a_i の和を求めればよい
- 絶対値が取れた
- 各aが、a_j – a_i の左側と右側に何回現れるか考える
- 左側 ... 自分より小さい添字とのペアなので、 i – 1 回
- 右側 ... 自分より大きい添字とのペアなので、N – i 回

C – Pair Distance 解説

- つまり … ？
- ソートした後、1 <= i <= N について a_i * ((i - 1) – (N – 1 - i)) を足せばよい
- O(N log N)

D – Concatenation 解説

- 概要
- “code” のように、先頭が(真に)最小の文字列を連結した文字列 S が与えられる
- 最小何個の文字列に分解できますか
- |S| <= 100000

D – Concatenation 解説

- 左端の文字をcとする
- c以下の文字が出てくるところまで伸ばして取ってよい
- また、c以下の文字が出てきたら区切る必要がある
- 文字列の先頭から見ていって、今保持している文字以下の文字が出てきたら答えを+1するだけでよい
- O(|S|)

E – Union 解説

- 概要
- 1 <= i <= T の整数がそれぞれ 0 個以上 a_i 個以下含まれる(重複を許す)集合を考える
- 同じ整数xを2つ削除して、x+1を1つ挿入するという操作を繰り返す
- 整数を1つだけに出来るような集合は何通りありますか

E – Union 解説

- まず、与えられた集合が条件を満たすかどうかの判定法を考える
- 集合のうち最小の整数をXとする
- Xより小さい整数はないので、XはX同士で消すしかない
- Xを消せるだけ消して、Xが余ったらNG
- 最小がX+1以上になったので、これを繰り返せば判定できる

E – Union 解説

- この判定法より、下から考えれば良さそうなことがわかる
- dp[i][j] := 整数 i まで考慮して、i が j 個余るような場合の数 とする
- この計算量は … ？
- 問題は j をどこまで考える必要があるか
- i-1 のものは高々300 * (1/2)個くらい、i-2 のものは300 * (1/2) * (1/2)個くらい…
- これはだいたい300
- (300(前から来たもの)+300(iの個数)) / 2 = 300
- これで時間計算量はO(T * max(a_i) ^ 2)になりました
- dpの更新をうまくやると O (T * max(a_i))にもなります

F – Coins on the tree 解説

- 概要
- N頂点の根付き木にM枚のコインを置き、ちょうどK回の操作を行う
- コインは根に置いた後、子に移していく
- 根も含めて、同じ頂点には同時に1枚しか存在できない
- (根に1枚目に置いたコインがある頂点,根に1枚目に置いたコインがある頂点,…,根に1枚目に置いたコインがある頂点)の列を辞書順で最小化

F – Coins on the tree 解説

- 設定が複雑
- 要は、後から置いたコインはそれより前に置いたコインを追い越せない
- M枚のコインの位置を決めたら、それが実現できるか
- v1, v2, …, vMとする
- v2 は v1 を根とする部分木に含まれてはいけない
- v3 は v2を根とする部分木 と v1 を根とする部分木 に含まれてはいけない
- …
- 逆にこれを満たせば、1枚目から順に一気に v_i まで移動させればよい
- 前から決めてよさそうなので辞書順と相性がよい

F – Coins on the tree 解説

- i-1 枚目まで置いて、次に i 枚目を置く場所を決める
- i 枚目を v に置いて良いか判定出来れば、vを小さい順に試せばよい
- 操作回数は、深さの和であることに注意 (根を深さ1とする)
- i-1 枚目までの深さの和が S で、vの深さがd_vであるとき、あと K – S – d_v 回の操作が出来る必要があり、また M – i 個の頂点が空いていなければならない
- v を根とする部分木の頂点を全て使えなくする
- 残っている頂点がM – i個以上
- 残っている頂点の、浅い方からM – i 個の和と深い方からM – i 個の和の間にK – S – d_v があれば帳尻合わせできる

F – Coins on the tree 解説

- 深さの条件の補足
- 木のまま考えると難しいが、深さの列を考えると、深い方から+1していくと全ての和が作れることがわかる
- 浅い方からの和と深い方からの和は、各深さの頂点の個数を持っておくと、バケツソートの容量でO(N)
- M枚のコインについてそれぞれO(N)頂点を試し、毎回チェックにO(N)かけるとO(M N^2)

G - Sum of Cards 解説

- 概要
- N枚のカードがあり、i 番目のカードの表にa_i, 裏にb_iが書いてある
- K種類以上が見えるようにカードの表裏を決めて、和を最大化する
- a_i, b_i はそれぞれ1,2…,Nの順列である
- N <= 5000

G - Sum of Cards 解説

- カードの作り方が重要
- 各整数は、表と裏にちょうど1回ずつ出てくる
- (表,裏)が(X,Y)のカードがある時、辺(X,Y)があるようなグラフを考える

G - Sum of Cards 解説

- 全部の頂点の次数が2であり、この無向グラフは輪っかの集まりとして表せる
- グラフの各辺について、どちらかの向きをつけ、各頂点について入次数の分だけ加算する、入次数が0でない頂点がK 箇所以上 と解釈できる
- 各輪っかについて、輪っかのどこかを基準として、dp[i] [j] [k] [l] := i番目の頂点まで見て、j種類の整数が現れ、k : 先頭が既に現れたかどうか、l : i+1 番目の頂点が現れたかどうか の場合の最大値 とおくとできる
- これでも出来るがもう少し考察
- 和は、(1からNまでの和) + (2回現れる頂点の和) – (0回現れる頂点の和)

G - Sum of Cards 解説

- 2回現れる頂点と0回現れる頂点に興味がある
- a -> b -> c となっている時a -> cに置き換えていくと、a <- b -> c <- d -> … となることがわかる
- 各輪っかについて、交互に i 個ずつ 2回現れる頂点と0回現れる頂点を選ぶことにすると循環にあまり気を使わなくてすみます
- どちらにせよ計算量は O(N^2)

H – Median Game 解説

- 概要
- 数列を使ってAliceとBobがゲームをする。
- Aliceから交互に、数列の先頭から1個以上消して、その和を記録する
- 和を記録したものたちに対して、Aliceは中央値を最大に、Bobは中央値を最小にするように振る舞う
- 2人が最適に行動した時の結果を求めてください
- 中央値は、偶数個の時は真ん中の右側としてある

H – Median Game 解説

- 中央値は結果の列全体に依存するので扱い辛い
- 突然ですが、Aliceは答えをX以上に、Bobは答えをX未満にしたいという設定を考える
- この結果が判定出来れば、答えを二分探索できるので嬉しい

H – Median Game 解説

- 中央値がX以上ということは、(X以上の個数) >= (X未満の個数) に出来るか
- 以下のようなdpを考える
- dp1(i) :=現在 Alice の番で、数列の先頭が i 番目のとき、それ以降でゲームした場合の (X以上の個数) – (X未満の個数) の最大値
- dp2(i) := 現在 Bob さんの番で、数列の先頭が i 番目のとき、それ以降でゲームした場合の (X未満の個数) – (X以上の個数) の最大値
- dp1(0) が 0以上かどうか
- これは 愚直な dpでも O(N^2) で出来る(区間の和は、dpしながら順番に足すか、累積和を使いましょう)
- 全部で O(N^2 log N)、おそらく O(N log^2 N)でも解けると思います