ABC 045 / ARC 061 解説

tozangezan

2016年9月11日

1 A: 台形

台形の面積は、上底をa、下底をb、高さをhとすると、(a+b)h/2 で求められます。この問題はh が偶数なので、答えは整数に収まるため、int 型で計算できます。

2 B: 3人でカードゲームイージー

ルールの通りにシミュレーションします。実装の方針としては、3人それぞれのカードの列を string 等で文字列にして持っておき、現在その文字列のうち何番目の文字がその人の持ってるカードのうち先頭にあるかを変数に持ちます。また、現在誰のターンであるかを表す変数も持っておきます。

それぞれのターンにおいては、まずそのターンの人がカードを全部使い切ったかを確認 し、そうでないならば自分のカードの位置を表すカウントを1増やしてから、文字に対応 する人のターンに移動します。

計算量は $O(|S_A| + |S_B| + |S_C|)$ なので、特に問題なく動きます。

3 C: たくさんの数式

与えられる文字列の長さの最大が10と小さいので、それぞれの文字と文字の間に'+'を入れるかどうかを全探索で試すことが出来ます。

何番目の数字の後に '+' を入れるかを全て決めたら、split 等の関数がある言語ならば、実際に新しく '+' を入れた文字列を作り、区切って計算することができます。そうでなくても、前から順に見ていって、「数字があるたびに現在の値を 10 倍して今書いてある数字だけ足す」「'+(', が出てきたら現在の値が '+(', の直前の整数に等しいので、それを答えに足す」「最後は '+(', が登場せずに数式が終わるので、そこは別に足す」とすると、新しく出来た式の値を求めることができます。

4 D: すぬけ君の塗り絵

盤面が大きいので、実際に 2 次元配列で塗ることをシミュレーションすることはできません。また、 N が 10^5 と大きいので、座標圧縮したとしても、メモリ使用量が $O(N^2)$ となり厳しいです。

ここで気づくべきポイントは、「一つの黒く塗られたマスが影響を与える範囲は、その周辺にあるいくつかの 3×3 領域しかない」ことです。すなわち、それぞれの塗られたマスに対し、これらの周辺の少しのマスにのみ何かの処理をするだけで計算できれば、高速に答えが求められることになります。

(i,j) を、上から i 行目で左から j 行目のマスを左上とする 3×3 の正方形を表すこと にします。例えば、上から p 行目で左から q 行目のマスを黒く塗ると、 (p-k,q-l)(k=0,1,2)(l=0,1,2) に含まれる黒いマスが 1 つ増えます。 (ただし、これらの正方形が盤面 からはみ出すことがあるので、そういうケースは除外します)

ということで、入力で与えられる点に対してこれらの正方形の左上のマスを全部列挙しておき (最大で 9N 個列挙される)、その列挙されたものの中で、それぞれの正方形が何回出現するかがカウントできればよいです。

これは、pair 型などで (i,j) をソートして前から順に同じ値が何回出てくるかを数えても良いですし、map 等のデータ構造を使ってもよいです。

最後に、黒いマス 0 個の個数は、全体 ((H-2)(W-2) 個) から 1 回以上黒いマスが出現した 3×3 領域の個数を引けば求められます。

計算量は、O(NlogN) となります。

5 E: すぬけ君の地下鉄旅行

(現在いる駅,最後に使った会社)を頂点に最短路問題を解くことで答えが求められます。この解法では、頂点数は高々O(M)です。なぜならば、それぞれの駅に到達するのに最後に使った会社の種類数は、最大でもその駅から出ている路線の個数です。この個数を全ての駅で合計しても、路線の個数の定数倍にしかなりません。

しかし、効率的な辺の張り方をしないと、辺の数が $O(M^2)$ 等になってしまい、制限時間内に実行が終わらなくなります。

例えば、次のような辺の張り方をすると辺の数は高々O(N+M)で済みます。

- 同じ会社の路線に乗り換えるケースに関しては、そのまま長さ 0 の辺を全ての路線 の目的地に対して張る
- ・違う会社の路線に乗り換えるケースの代わりとして、(現在いる駅,−1)という頂点 に移動できるように長さ0の辺を張る。(改札から外に出るイメージ)
- (現在いる駅, -1) の頂点から全ての会社の路線に対して、目的地に長さ1の辺を張る (改札の中に入るイメージ、この時点で新たに料金がかかる)。
- スタートは、(1,-1)の頂点。ゴールは、(N,-1)の頂点。

あとは、Dijkstra 法等で十分早い時間で答えを求めることができます。

6 F: 3人でカードゲーム

6.1 部分点解法

この問題では、3人がそれぞれカードを N, M, K 枚ずつ持っている設定ですが、誰かがカードを使い果たしてゲームに勝利するまでの動作に関しては、「誰かがカードを出す」操作の繰り返しだと考えることができます。そこで 3 つのカード列を考える代わりに、1 つの列にまとめて考えることができます。

この状況下でAさんが勝つ必要十分条件は、一列にしたカード列において、Bが M回以下、Cが K回以下しか出ていないときにBが、回出ることです。B0回目のB0が、全体で何ターン目にあたるかを全て試します。

i ターン目で N 回目の $\mathbf a$ が出るとき、条件を満たす配置の個数は、「i ターンで使われなかった余りのカードの選び方 $3^{N+M+K-i}$ 通り」「i-1 ターン目以前のうち $\mathbf a$ が出る N-1 回を選ぶ方法の数 i-1 C_{N-1} 通り」「 $\mathbf a$ 以外が出る i-N ターンを $\mathbf b$ と $\mathbf c$ で埋める方法の数」の積です。

最初の2つは簡単に計算することができます。最後の「a 以外が出るi-N ターンを b と c で埋める方法の数」は、 $_{i-N}C_k$ $(max(0,i-N-c)\leq k\leq min(i-N,b))$ の総和です。これらを全部一つずつ計算すると、ここだけで $O((M+K)^2)$ となり、部分点は取れますが、満点は取れません。

6.2 満点解法

さっきの二項係数を足す連続する範囲を全てのiについてパスカルの三角形上に書いてみると、このような形になります。

*

**

.**...

これらを、上の段から一段ずつまとめて計算していきます。最初、一番上の段を見ていて、この範囲における求めたい合計は1です。

ここから降りていくとき、左下、右下ともに新たに追加されるうちは、その段の合計は、 一つ上の段の合計の 2 倍となります。

左下、右下の片方しか追加されなくなったとき、その段の合計は、一つ上の段の求める 範囲の合計の 2 倍から、追加されないほうの端に影響を与える、一つ上の段で求める範囲 の中にあった点の値を引きます。 左下、右下のどちらも追加されなくなったとき、その段の合計は、一つ上の段の求める 範囲の合計の2倍から、追加されない両端に影響を与える、一つ上の段で求める範囲の中 にあった点の値を両方引きます。

このようにまとめて段ごとに計算することで、計算量は O(M+K) となり、満点を取ることができます。