Mujin Programming Challenge 2018 解説

DEGwer,satashun

2018/08/04

A: コンテスト名

文字列の長さが 5 以上かどうかと、1...5 文字目がそれぞれ正しいかどうかチェックすれば良いです。

B: セキュリティ

文字列を前からシミュレーションして、+ なら A に 1 を足し、-なら 1 を引いて毎回 0 かどうかを判定すると、$O(|S|)$ で解くことが出来ます。初めと終わりに注意してください。

C: 右折

向きを 90 度変えるマスを固定して考えることにしましょう。あるマスで向きを 90 度変えるような始点と終点の組の個数を高速に求められれば、この問題を解くことができます。これは、そのマスから上下左右にそれぞれ (そのマスを含まずに)u, d, l, r マスの障害物のないマスが続くとすれば、$(u + d) \times (l + r)$ 個となります。この u, d, l, r はそれぞれ上下左右から順にマス目を見ていくことで $O(NM)$ 時間で求められるので、$O(NM)$ 時間でこの問題を解くことができます。

D: うほょじご

$0 \leq x, y \leq 999$ なる (x, y) の組に対して頂点を 1 個ずつ用意し、$1 \leq x, y$ なる (x, y) に対し、頂点 (x, y) から (x, y) に操作を行ってできる組に対応する頂点に辺を張ったグラフを考えたとき、求める個数は $1 \leq x \leq N, 1 \leq y \leq M$ なる (x, y) であって対応する頂点から辺に沿って進み続けたときに無限ループするような頂点の個数です。逆に、無限ループを起こさないような頂点の個数を数え、全体から引くことを考えましょう。

これは、上記のグラフで辺を逆向きに張り、$(0, x)$ または $(x, 0)$ の形をした頂点から行ける (与えられた範囲に収まる) 頂点の個数を数えることで実現できます。

E: 迷路

最初に、愚直な解法として dist2$[i][j][k]$: マス (i, j) に時間 mod K が k となるような時間で到達するような最短の時間 とおき、幅優先探索する解法が考えられますが、このままでは計算量がかかりすぎます。しかし

各マスに最短で行った後そのマスで待つことが出来るため、$\mathrm{dist}[i][j]$: マス (i, j) に到達する最短の時間とおいても正しく計算することがわかります。このマスから隣のマスへ移動するためには、$k = \mathrm{dist}[i][j] \bmod K$ として各方向の文字が次に出てくるところまで待つ必要があり、このコストが高速に求まると Dijkstra 法を用いて答えを計算することが出来ます。このコストは例えば事前に d の後ろにもう 1 つ同じものを足し、後ろから dp しておくことで $O(K)$ 時間で全て計算しておくことが出来ます。よって時間計算量 $O(NMlogNM)$ でこの問題が解けました。

F: チーム分け

まず、a_i が小さい人ほど所属出来るグループの条件が厳しいです。$a_k = i$ となる人数を各 i について求めておき、c_i とおきます。サイズの異なるグループ同士は明らかに区別出来るので、グループのサイズの大きさごとにまとめて、大きい方からグループを増やすことを考えます。$\mathrm{DP}[i][j]$ を、a が i 以上の人の中から、サイズ i 以上のグループまでを作って、j 人どのグループにも属さずに余っているようなグループ分けの場合の数とおきます。$\mathrm{DP}[i+1][j]$ からの遷移を考えます。i 人のグループを作る際に割り振ることの出来る人は $j + c_i (= num)$ 人います。i 人のグループを k 個作る場合、$\mathrm{DP}[i][num - ki]$ に寄与し、その係数は $\frac{num!}{(num-ki)! * (i!)^k * k!}$ です。各 i についてサイズ i のグループは高々 $\frac{N}{i}$ 個しか作ることが出来ないため $\sum_{i=1}^{N} \frac{N}{i}$ が $O(NlogN)$ であることを考慮して、時間計算量 $O(N^2 logN)$ でこの問題を解くことができます。

G: 移動

与えられた 3 本のベクトルを v_1, v_2, v_3 とします。もし $0 \leq a, b, c$ かつ $a + b + c \leq K$ なる (a, b, c) に対し、$av_1 + bv_2 + cv_3$ がすべて異なれば、答えは簡単に求めることができます。

そうでない場合に、重複分を引くことを考えましょう。もし $av_1 + bv_2 + cv_3 = a'v_1 + b'v_2 + c'v_3$ なら、$(a - a')v_1 + (b - b')v_2 + (c - c')v_3 = 0$ です。また、問題文の条件より、このような $((a - a'), (b - b'), (c - c'))$ の組は定数倍を除いてただ 1 つ存在することが分かります。

よって、求める重複分は、このような組を (x, y, z) として (これは線型方程式を解くことで求まる)、$0 \leq a, b, c$ かつ $a + b + c \leq K$ かつ $0 \leq a + x, b + y, c + z$ かつ $(a + x) + (b + y) + (c + z) \leq K$ なる (a, b, c) たちの個数であり、これは簡単に求めることができます。よってこの問題が解けました。

H: タイル張り

最初に、黒マスの位置がすべて決まっているときに、条件を満たすタイル張りが存在するかどうかを判定する方法を考えましょう。これは、$\mathrm{DP}[i][m]$: 左から i 列見て、横向きに置いたタイルが左の列にはみ出している場所の集合が m (としてビットマスクで表されるようなもの) であるようなタイルの置き方が存在するかどうか という bitDP で判定できます。

さて、黒マスの位置が決まっていない場合を考えましょう。これは、「左から i 列見たとき $\mathrm{DP}[i][m]$ が真となるような m の集合 (すなわち、列のべき集合の部分集合)」をキーとして bitDP を行えばよいです。ただし、キーとしてすべての集合を見る場合、そのキーの個数は 2^{2^H} となり、とても現実的なサイズではありません。

しかし、そのようなビットマスク m の集合として元の DP の途中に現れるようなものの個数は、2^{2^H} より

はるかに小さいことが分かります。実際、それに属する集合の元数の偶奇はすべて等しいなど、この個数を大幅に小さく評価できる基準があります。

実際に数えてみると、$H = 5$ でそのような集合はわずか 91 個しかないので、行列累乗を用いてこの問題を解くことができます。