ARC051解説

A問題 塗り絵

この問題は,2次元平面で円と長方形が与えられるので

- 円が完全に長方形に内包されているか
- 長方形が完全に円に内包されているか

の2つを判別すればよい.

円が完全に長方形に内包されているか

- x2 <= x1-r
- x1+r <= x3
- y2 <= y1-r
- y1+r <= y3

上記4つが必要十分条件となる.

長方形が完全に円に内包されているか

これは,長方形の4つの頂点が全て円に内包されているかを調べれば良い.

B問題 互除法

部分点は乱数を使い,いろんな入力をたくさん試せば作れる.

満点はフィボナッチ数列を使うと良い.

- F_1 = F_2 = 1
- F_k = F_{k-1} + F_{k-2} (k >= 3)

フィボナッチ数列とは上記の規則によって生成される数列であり,gcd(F_i+1, F_i)を呼ぶと,そこからgcd(F_i, F_i-1)が呼ばれることがわかる. これを繰り返すと, gcd(F_2, F_1), つまりgcd(1, 1)が呼ばれ, そこからgcd(1, 0)を呼び, 関数が終了する.

よってgcd(F_{K+1}, F_K)を呼ぶとこのプログラムの出力はKになることがわかる.更に、F_41は267914296であるので, 10^9という制約を満たせる.

C問題 掛け算

10^9倍するという動作を10^9回ぐらい行うので, 普通に計算することはできないが, 整数たちの値が十分近ければ, 周期的に掛け算をするようになる.

整数たちをa_1, a_2, ..., a_N(a_1 <= a_2 <= ... <= a_N)とする. 実は a_1 * A >= a_N とすると, a_1, a_2, ..., a_N, a_1, a_2, ..., a_N, a_1, a_2, ..., a_N, ... と掛け算をしていくようになる.

なので, a_1 * A >= a_N となるまでは愚直にシミュレーションをして, そのあとは何回A倍されるかというのを求めれば良い.ただしK=1の場合はコーナーケースとなるので注意すること.

計算量は, 最初のシミュレーションで行う掛け算が高々O(N * log(max a_i))回ぐらいなので, 最小値を探すのにO(N)かけてもO(N^2 * log(max a_i)). そして, 何回A倍されるかを求めた後の高速累乗でO(N * logA).

よって十分間に合う.

D問題 長方形

まず, 長方形を考えたときに値の総和はいくつになるか考える.列の幅をw, 数列の和をXとして, 行の幅をh, 数列の和をYとすると, wY+hXとなることがわかる.

よって列の幅を固定したら, 列の数列の和が最大となるような選び方のみ考えれば良い. 行についても同様.

つまり

- max_w_sum[i][j] := 左からi番目以内までで, 幅jとなるように選んだ時の数列の和のmax
- max_h_sum[i][j] := 上からi番目以内までで, 幅jとなるように選んだ時の数列の和のmax

という配列を作っておけば, クエリは以下のように言い換えられる.

- A, Bが与えられるので, w×max_h_sum[B][h] + h×max_w_sum[A][w]の最大値を求める.(1 <= w <= A, 1 <= h <= B)

更に, wを固定して考えてみると,

- w×(max_h_sum[B][h] + h×max_w_sum[A][w]/w)の最大値を求める.(1 <= h <= B)

という問題になるが, これは以下のように言い換えられる.

- y = h×x + max_h_sum[B][h]という直線が1 <= h <= Bそれぞれについて B本ある. xにmax_w_sum[A][w]/wを代入した時の, y座標の最大値を求める.

これは, convex hull trickを使うと初期化O(B)クエリO(1)で求めることができる.ただし, 事前にmax_w_sum[A][w]/wをsortしておく必要があり, それにO(AlogA)かかる.

よってクエリあたりの計算量はO(B+AlogA) ⊂ O(H+WlogW).

よって計算量はO(Q(H+WlogW))で, 間に合う.