蟻と角砂糖

解説担当：川崎 理玖

問題概要

- 直線上に蟻と角砂糖が置かれている
- 各蟻を距離 L 以内の角砂糖に向かわせる
- 食べられる角砂糖の個数を最大化
- クエリで蟻や角砂糖が追加される

問題概要

- 距離 L 以下の蟻と角砂糖がマッチングできるとして，最大マッチングを求めればいい

小課題1

- 各クエリごとにO(N)で答えを求める．（N はそれまでに処理したクエリの数）

- 蟻，角砂糖それぞれについて，（位置，個数）のリストを位置の昇順に持っておく．

小課題1

- 座標最小の蟻と座標最小の角砂糖に注目
- マッチングできるならする
- マッチングできないなら，座標の小さい方は以降無視
- あとで損することがないため，上の貪欲が最適

小課題1

- 普通に実装すれば O(N)
- 全体 O(Q^2)
- 6 点
- ちなみに余計な log をつけても通ります

小課題2

- 蟻と角砂糖の座標が偶奇で分かれている
- L=1 なので，ちょうど隣のものだけペアにできる

小課題2

- Hall の定理を使うとわかりやすくなる
- IOI Syllabus 上での Hall の定理の扱いは（多分）グレー
- JOI春合宿ではこうした問題も当然のように出題

小課題2

- この問題に適用してみる
- 今存在する蟻の総数をＡとおく
- 蟻の部分集合Ｓを選び，Ｓとマッチング可能な角砂糖の集合をＴとおく

小課題2

- マッチングサイズが A-|S|+|T| 以下であるのは自明
- 実は，A-|S|+|T| の最小値が最大マッチングのサイズに一致（Hall の定理を使う）
- Hall を経由しなくても示せる（小課題1の貪欲をよく観察する）ので実はこれが想定かも

小課題2

- 同じ座標にいる蟻は，すべて選ぶか選ばないかの2択
- 各座標 1,3,5,… について，その座標にいる蟻を選ぶかどうか決める
- 便利のため，蟻が 0 匹の座標を選んでもいい

小課題2

- 座標 1,3,5,... を選ぶかどうかを 0,1 列 x_0,x_1,... で表すことにする
- x_i を決めたときの A-|S|+|T| への寄与：

  x_i が 1 なら (座標 i*2-1 にいる蟻の数)*(-1)

  x_i か x_{i+1} が 1 なら (座標 i*2 にある角砂糖の個数)

小課題2

- 全体のスコアが局所的なスコアの和で表わせた
- Segment Tree に乗せてみる

小課題2

- 各ノードは，そのノード内だけ見たときの -|S|+|T| の min を持つ
- ノードのマージを考えると，ノード内の左端，右端を選んだかどうかで 4 通り持っておくべき

小課題2

- 区間 [a,b],[b+1,c] をマージする際は，b,b+1 の間にある角砂糖の個数を参照して適宜計算
- クエリあたり O(log Q)，全体 O(Q log Q)
- 16 点

小課題3,4

- L の値が大きいため，小課題 2 のようにスコアを局所的なスコアの和に分解できない
- と思いきや，実はできる

小課題3,4

- 小課題2と同様，蟻がいる座標を昇順にならべ，それを使う/使わないを x_i で表す．
- ある角砂糖の寄与は，以下の形になる．
- x_l,x_{l+1},…,x_r のうち 1 つ以上が 1 なら，(+1) の寄与

小課題3,4

- これを次のように変形する
- x_i が 1 なら (+1) の寄与 ($l<=i<=r$)
- x_i と x_{i+1} が両方 1 なら (-1) の寄与 ($l<=i<=r-1$)

小課題3,4

- 変形により，寄与が本来より小さくなることはない
- 最終的に最小化するので，不正に小さい解を出力することはない
- 最適解をちゃんと達成できることを示す

小課題3,4

- $l \le a < b \le r$ に対して，$x_a=1, x_b=1$ であり，$x_i=0$ $(a<i<b)$ だったとする
- このとき，$a<i<b$ を満たすすべての i で x_i を 1 に変えても問題ない

小課題3,4

- a<i<b を満たす何らかの i に対して，x_i に作用するような角砂糖を考える
- マッチングの条件を考えると，この角砂糖は必ず x_a,x_b の少なくとも一方に作用している
- よって，前ページの変形を行っても損しない

小課題3,4

- 結局，どの角砂糖に対しても，x_l,x_{l+1},...,x_r のうちある連続区間が 1 になるという形になる
- つまり，角砂糖のスコアへの寄与は正しく求まる

小課題3

- 角砂糖の位置が予めすべて与えられているなら，蟻の更新は小課題2と全く同様に行える
- （問題では蟻→角砂糖の順番なので役割を入れかえる）
- 全体で O(Q log Q) となり，26 点

小課題4

- 小課題2,3で使った Segment Tree に更に操作を追加する
- 角砂糖の追加を区間加算クエリの要領で処理する

小課題4

- 角砂糖が x_l,x_{l+1},…,x_r に作用する時，区間 [l,r] に対してクエリを呼ぶ
- 今，あるノード全体にこのクエリが飛んだとする

小課題4

- このノードは自分以下での -|S|+|T| の min を保持している
- 角砂糖を W 個追加するなら，この min はすべて (+W) されるはず
- ただし，|S|>0 でないときは違う

小課題4

- ノードが持つデータを，-|S|+|T| の min (ただし，|S|>0 であると仮定) としておけばよい

小課題4

- ノードのマージの際，マージする 2 つのノードの境界に作用する角砂糖の個数を知る必要がある
- これを別個にBIT で管理すると余計な log がつく（が高速 BIT なら通るかもしれない）

小課題4

- 実は同じ Segment Tree 内で処理できる
- 各ノードに，「その2つの子の境界に作用する角砂糖の個数」を記録することにする
- 区間加算 Segment Tree と同様に，遅延評価を用いると効率的に更新できる

小課題4

- 計算量はクエリあたりO(log Q) になり，全体で O(Q log Q)
- 満点

得点分布