JAG夏合宿2013 day4-D
Removing Magical Tiles

原案:-
解答:八森、宮村
スライド:宮村

問題概要

- 平行な2直線上に辺を持つ台形がn個与えられる。
- 頂点数nのグラフを考える。

  i番目とj番目の頂点が隣接する

  ⇔ i番目とj番目の台形が交わる
- このグラフでの最小クリーク分割（被覆）を求めよ。

- $n \leqq 10^5$
- 台形の頂点が重なることはない。

半順序を入れる

- 2つの台形が互いに交わらず、台形uが台形vの左側にあるとき$u < v$であると定義する。これは推移的で、つまり半順序になっている。

- グラフとの対応を考えると、

  $(u,v) \in E \quad \Leftrightarrow \quad$ $u<v$でも$v<u$でもない

- つまり、補グラフが推移閉包をとったDAG（正確にはcomparability graph）。

鎖と反鎖

- 半順序の部分集合で、任意の異なる2つの元が比較可能(不可能)であるとき鎖（反鎖）であるという。
- 本問題におけるグラフと半順序の対応は、

  クリーク ⇔ 反鎖

  独立集合 ⇔ 鎖
- 求めるものは|最小クリーク分割|=|最小反鎖分割|

クリーク分割と独立集合

- 半順序集合ではMirskyの定理が成り立つ。つまり
  |最小反鎖分割| = |最大鎖|
- 右辺が左辺の下界であることと、等号を満たすような反鎖分割が簡単に構成できることから示せる。

- したがって、求めたい|最小クリーク分割|は
  |最大鎖|=|最大独立集合|と等しい。
- |最大鎖|とはつまり、単なるDAGの最長パスである。

動的計画法

- 結局求めたいものはDAGの最長パスなのでDP で計算可能。
- opt[u] := max(|uが最大である鎖|)　とおく。
- opt[u] = 1　(uが極小)

  1 + max(opt[v]; v < u)　(それ以外)

- 番兵を置いておけば下の式でまとめられる。
- これをそのまま計算するとO(n^2)でTLEとなる。

平面走査

- 以下の2つをサポートしたrmqを0初期化して用意しておく。
  rmq.get(x) : 区間[-∞,x)での最大値を返す
  rmq.update(x, v) : 点xの値をvに更新する

- xLowerの昇順方向に平面走査する。
- for xLower[i,j] in xLower[*,*]: ###昇順に舐める
  if j == 1:
    opt[i] = 1 + rmq.get(xUpper[i,1])
  else: ### j == 2
    rmq.update(xUpper[i,2], opt[i])
- 最終的にrmq.get(∞)が答。計算量はO(n*log n)。

実装

- この問題の性質から、RMQの実装はsegment treeでなくbinary indexed treeでも可（実装楽）。
- RMQを使わず値の追加、削除をサポートした平衡二分木（std::mapなどで可）を用いる方法もある。こちらの方が実装量を減らせる。
- LISのような配列と二分探索を用いた解法もある。これだと実装量はさらに減る。

参考資料

- 最小クリーク分割だけでなく最大重みクリークや彩色数などの問題が同様の平面走査で解ける。

- Stefan Felsner, Rudolf Muller, and Lorenz Wernisch. Trapezoid graphs and generalizations, geometry and algorithms.
- http://www.graphclasses.org/classes/gc_59.html