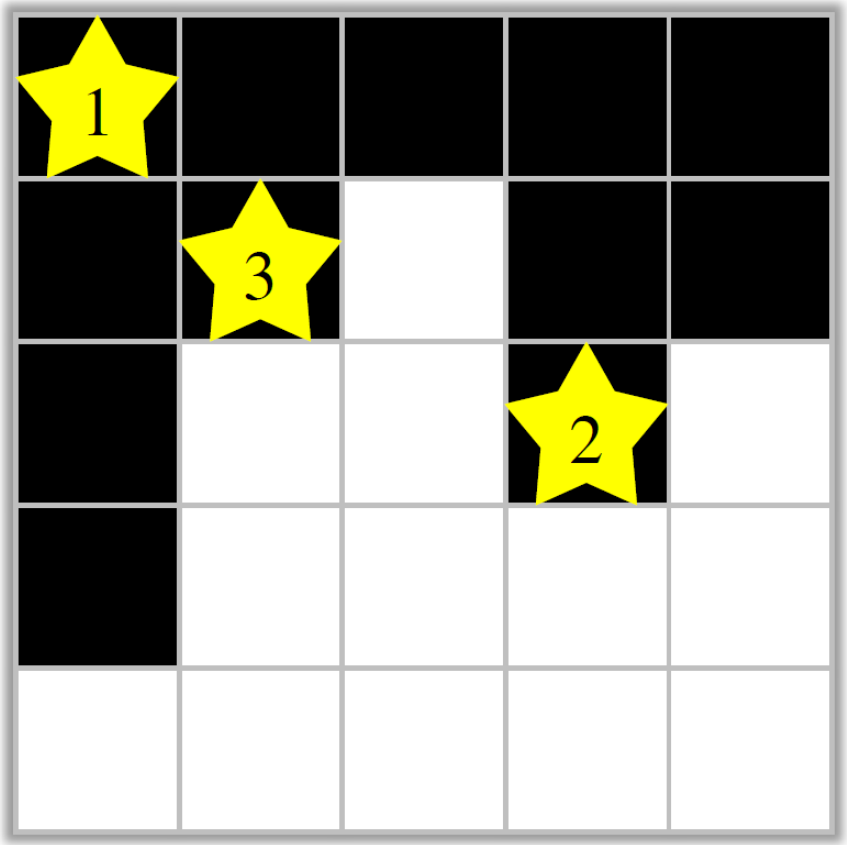
Constellation 3

高谷 悠太

問題概要

- ビルの高さ$A_1, \ldots, A_N$, 星$(X_1, Y_1), \ldots, (X_M, Y_M)$が与えられる
- 各星を消すのにかかるコスト$C_1, \ldots, C_M$が定まっている
- 以下を満たすように星をいくつか消すときの最小コスト
  - どの黒い長方形内にも高々1つしか星がない

サンプル



はじめに

- 星を消すのは面倒なので星を選ぶ問題にする
- どの黒い長方形内からも高々一つしか選べない

Subtask 1

- 動的計画法をする
- $dp[l][r][h] :=$ ($[l, r] \times [0, h]$から選べる最大コスト)
- 高さhの星を選ぶ or 高さ$h-1$に下がる
- 愚直にやると状態量3乗、遷移がたくさんで0点

Subtask 1

- 実は状態数は少ない

- 遷移過程で出てくる(l, r)として考えられるものはN個しかない

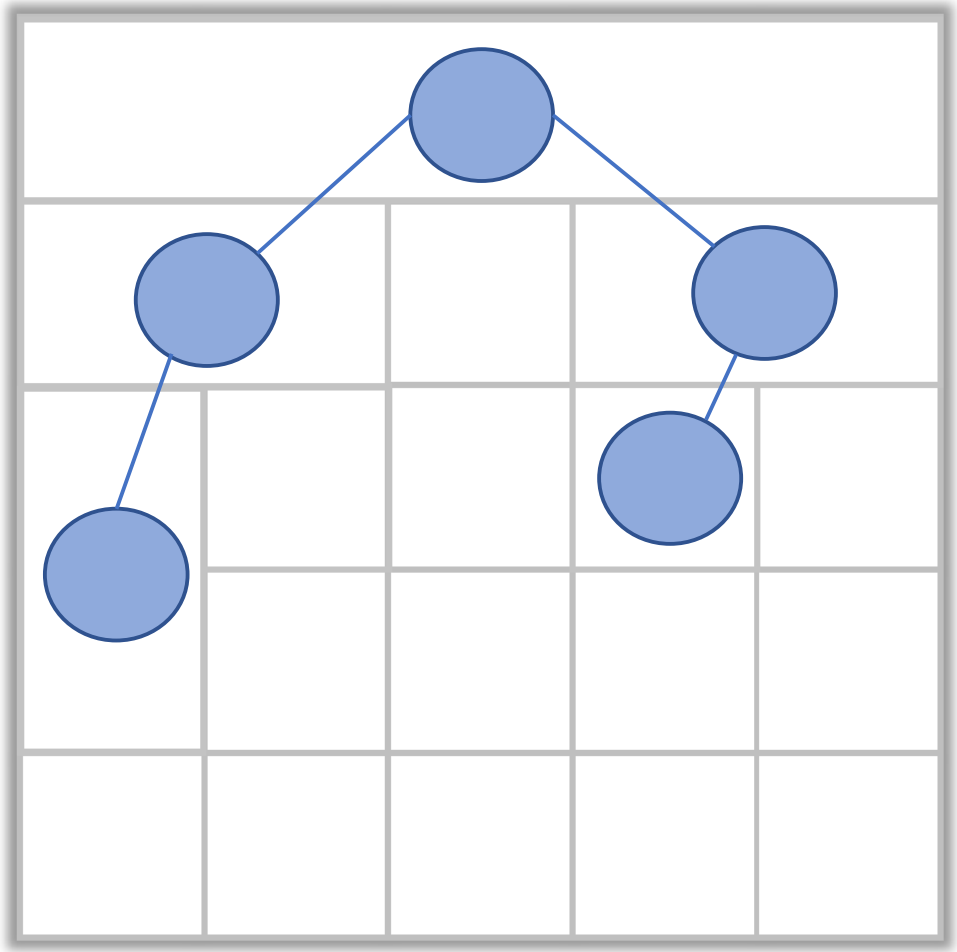
Subtask 1

- 実は状態数は少ない
- 遷移過程で出てくる(l, r)として考えられるものはN個しかない
- Cartesian Treeの頂点に現れる区間だけだね

Cartesian Tree の説明

- 最大値をとるindexを根にする
- 左右に分割する
- 左右で再帰的に木を作り、その根の親を全体の根につなげる

Cartesian Tree の説明 – サンプル



Subtask 2

- 必要な$[l, r]$の個数がN個だった
- 今度はhの候補が$[l, r]$に対して一意に定まる

Subtask 2

- 必要な$[l,r]$の個数がN個だった

- 今度はhの候補が$[l,r]$に対して一意に定まる
  - $[l,r]$に対応するCartesian Tree の頂点に対応する高さ

Subtask 2

- 必要な$[l,r]$の個数がN個だった

- 今度はhの候補が$[l,r]$に対して一意に定まる
  - $[l,r]$に対応するCartesian Tree の頂点に対応する高さ

- これでオーダーが一つ落ちる
  - 星についても、それが含まれる頂点でのみ遷移を行うことにする

満点

- 選べる星の集合の条件をちゃんと言い換える必要がある

- 「どの黒い長方形にも高々1つ」をCartesian Treeの言葉を用いて表す

満点

- まず各星には、それが含まれる頂点を対応させる
- その頂点は、星のx座標に対応する頂点の先祖にあたる

満点

- まず各星には、それが含まれる頂点を対応させる
- その頂点は、星のx座標に対応する頂点の先祖にあたる
- そこで、星に対応する頂点とx座標に対応する頂点を結ぶパスを考える

満点

- 星に対応する頂点とx座標に対応する頂点を結ぶパスを考える

- 「どの黒い長方形にも高々1つ」は
  「対応するパスがどの2つも交わらない」と言い換えられる

満点

- 結局次のような問題になる
  - 根方向のみに伸びるパスがいくつか与えられる
  - 各パスにはスコアが定まっている
  - どの2つも交わらないようにパスをいくつか選ぶ
  - 選んだパスのスコアの合計の最大値は？

満点

- 結局次のような問題になる
  - 根方向のみに伸びるパスがいくつか与えられる
  - 各パスにはスコアが定まっている
  - どの2つも交わらないようにパスをいくつか選ぶ
  - 選んだパスのスコアの合計の最大値は？

- これは木dpをすれば解ける

満点

- 具体的には、各頂点について、その頂点を根とする部分木に含まれるパスのみを考えて同じ問題を解く
- この答えをbottom-upの順に計算していく
- 子孫での答えが分かっているときに、根での答えを求めたい

満点

- まず、根を含むパスを選ばないとき
- この場合の答えは、直接の子どもでの答えを足し合わせたものになる
- 問題なのは、根を含むパスを選ぶ場合

満点

- 選ぶパスを1つ固定する
- そのときに得られるスコアの最大値はそのパスのスコアに

  そのパスを除いて得られる根付き森における根での答えの総和

  を加えたもの

- これをデータ構造を用いて計算したい

満点

- 各頂点で、直接の子での答えの総和を計算しておく

- その値をパス上の頂点に対して足し合わせたものから、根を除いたパス上の頂点での答えを足し合わせたものの差が、先ほど求めたかった値となる

満点

- 各頂点で、直接の子での答えの総和を計算しておく

- その値をパス上の頂点に対して足し合わせたものから、根を除いたパス上の頂点での答えを足し合わせたものの差が、先ほど求めたかった値となる

- 結局パス上に書かれた値の総和を求める問題が解ければよい
- これは解けるね