Port Facility 解説

DEGwer

問題概要

- 荷物がN個ある
- スタックが2個ある
- i個目の荷物は、時刻Aiにどちらかのスタックの一番上に積まれ、時刻Biにどちらかのスタックの一番上から取り出されるようにしたい
- そのような積み方は何通りあるか?

- N≦10^6

小課題1

- N<=20
- O(2^N N)で全探索してください
- やるだけ

ちょっと考察

- 時系列を数直線とみなし、各荷物を区間と考える
- 区間[Ai,Bi]と[Aj,Bj]を同じスタックに積める条件は?

ちょっと考察

- 区間[Ai,Bi]と[Aj,Bj]を同じスタックに積める条件は?

- ①二つの区間が重ならない場合

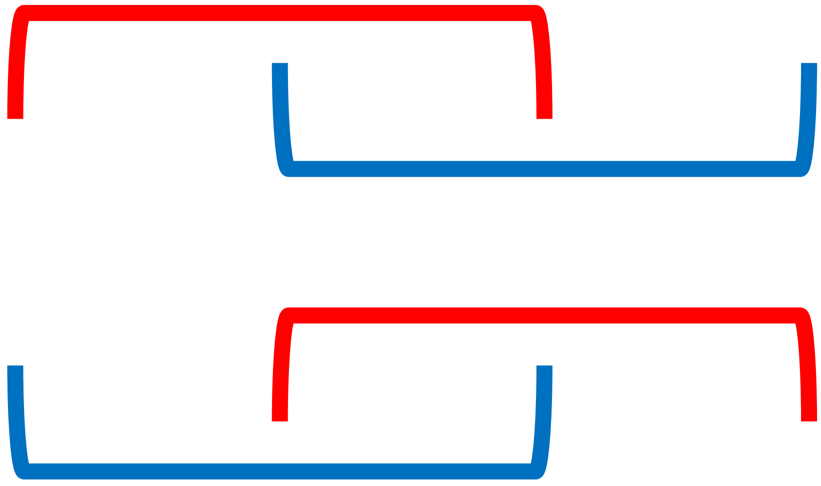
ちょっと考察

- 区間[Ai,Bi]と[Aj,Bj]を同じスタックに積める条件は?
- ①二つの区間が重ならない場合

- ②片方が片方に完全に含まれる場合

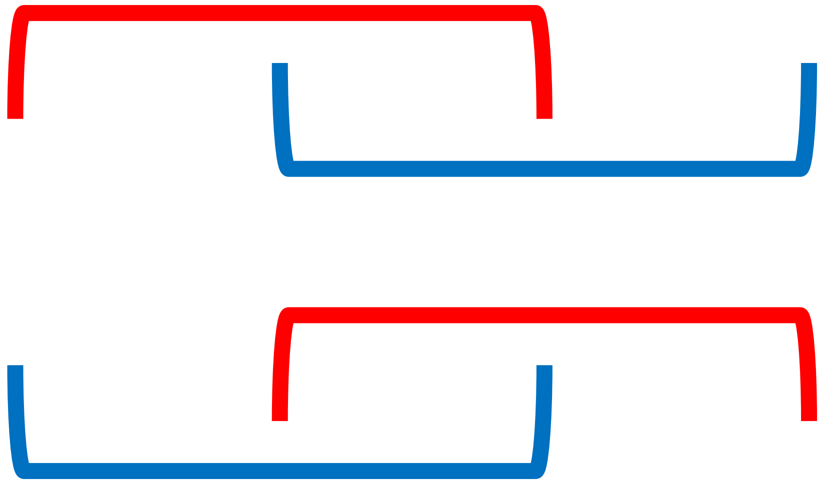
ちょっと考察

- 逆に、どちらでもない場合は同じスタックに積めない



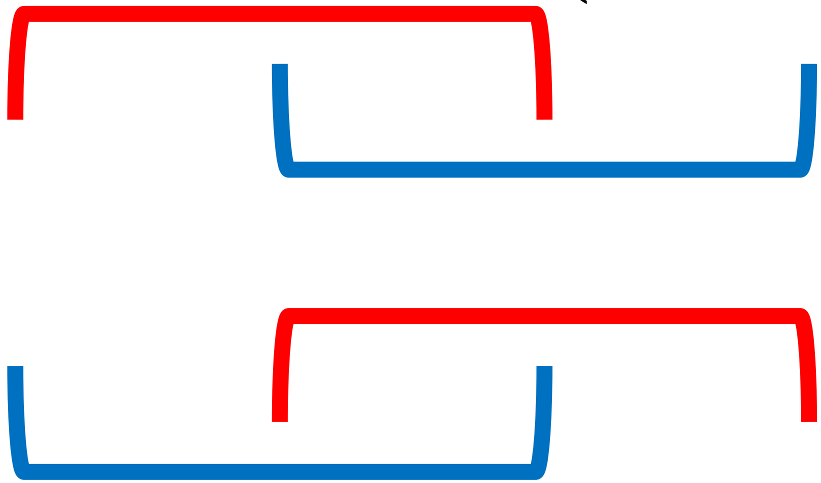
ちょっと考察

- 二つのスタックをA,Bとして、スタックAに積む荷物とスタックBに積む荷物に分けたとき、同じスタックに積む荷物が、こういう位置関係にあってはならない



ちょっと考察

- 荷物を頂点とし、こういう位置関係の二つの区間の間すべてに辺を張ったグラフの二部グラフ判定をすればよい
  - 二部グラフでないなら、答えは0
  - 二部グラフなら、答えは2^(連結成分数)



小課題2

- N<=2000
- 愚直に辺を張って二部グラフ判定をしましょう
- O(N^2)

小課題3,4

- N<=10^5, 10^6
- ここからは小課題2の高速化です
- いろいろやり方がある気がしますが、小課題3はO(Nlog^2 N)、小課題4はO(Nlog N)です

- 小課題4の実装をさぼったりすると小課題3になったりします
- 満点になりうる解法をいくつか紹介しますが、それ以外にもあるかもしれません

満点解法①

- 二部グラフ判定をする幅/深さ優先探索を考える
- 頂点数O(N), 辺数O(N^2)で、全部の辺を見ると人生終了

- とりあえずまず、グラフを連結成分に分解することを考える
  - 二部グラフ判定はあとで

満点解法①

- 辺を全部見るのは無駄
  - 既に訪れた頂点に行く辺はそもそも見なくてよい
- 既に見た頂点に行く辺は、見る前に削除されているようにしたい
- とはいえ、いちいち辺を一個ずつ削除すると、それもO(N^2)
- 辺をまとめて消したりしたい

満点解法①

- 各区間に「隣接する」区間はどのような区間か?
- 今見ている区間を[L,R]として、

- Lより前で始まって[L,R]間で終わる区間
- [L,R]間で始まってRより後で終わる区間

- 対称なので、片方だけ考えることにする

満点解法①

- Lより前で始まって[L,R]間で終わる区間
- これをどうやって探すか?

満点解法①

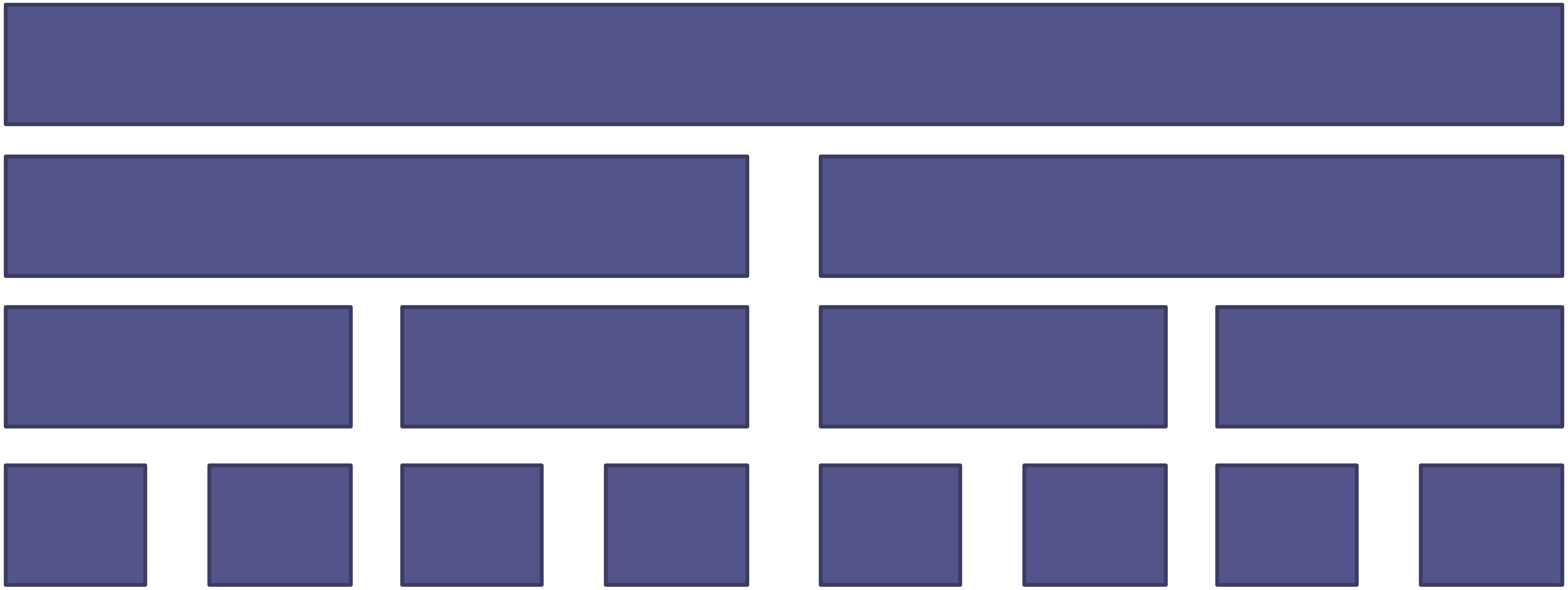
- Lより前で始まって[L,R]間で終わる区間
- これをどうやって探すか?
- segment tree

Segment tree

- Lより前で始まって[L,R]間で終わる区間
- これを高速に列挙するには?

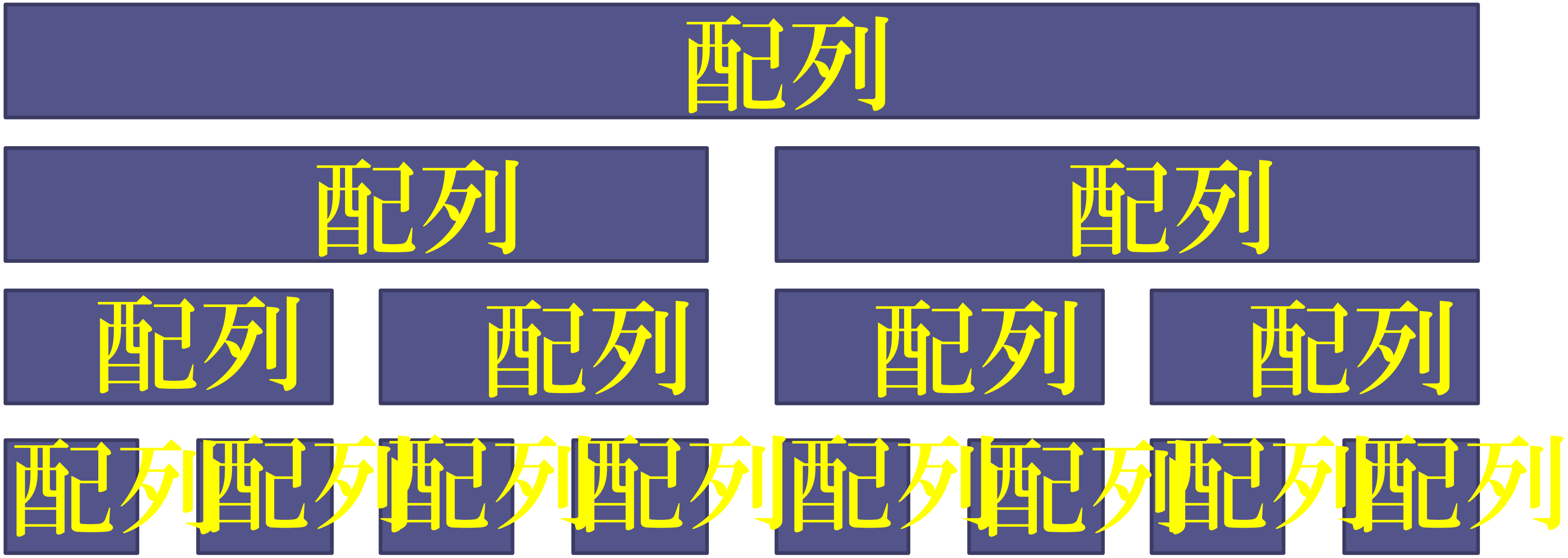
Segment tree

- 各ノードに配列を持っておく



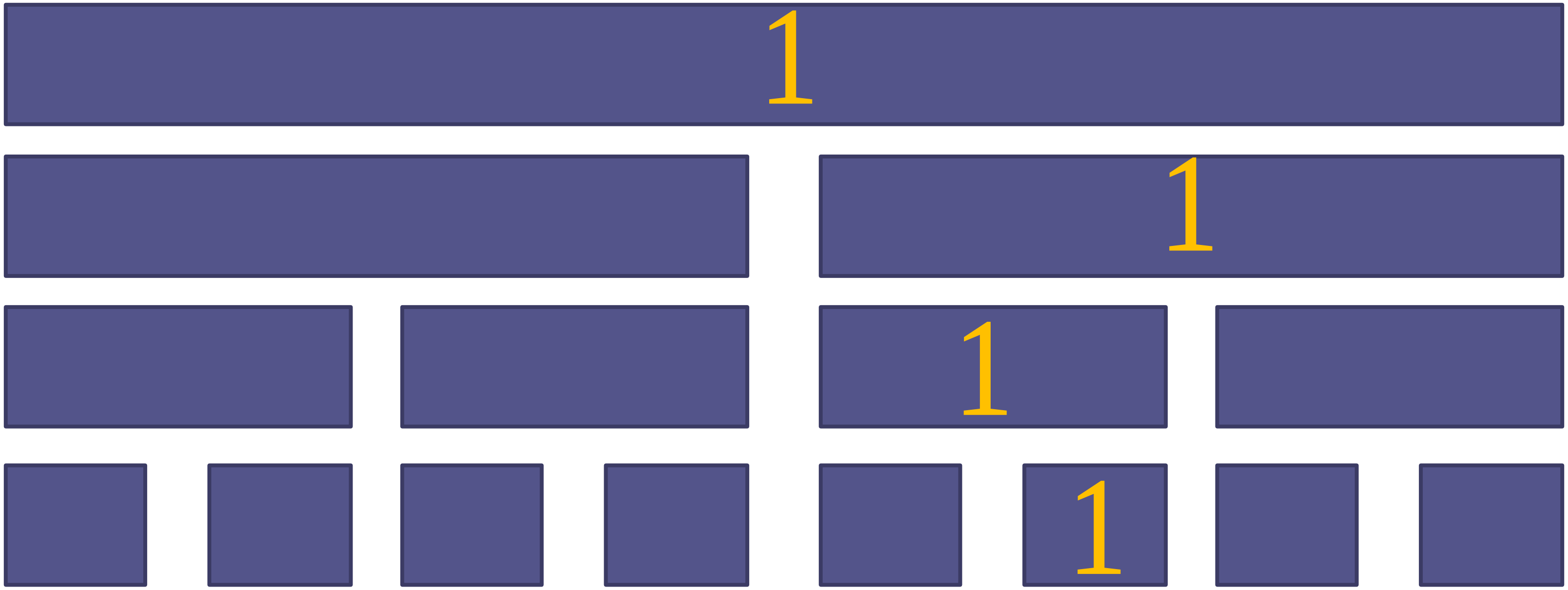
Segment tree

- 各ノードに配列を持っておく



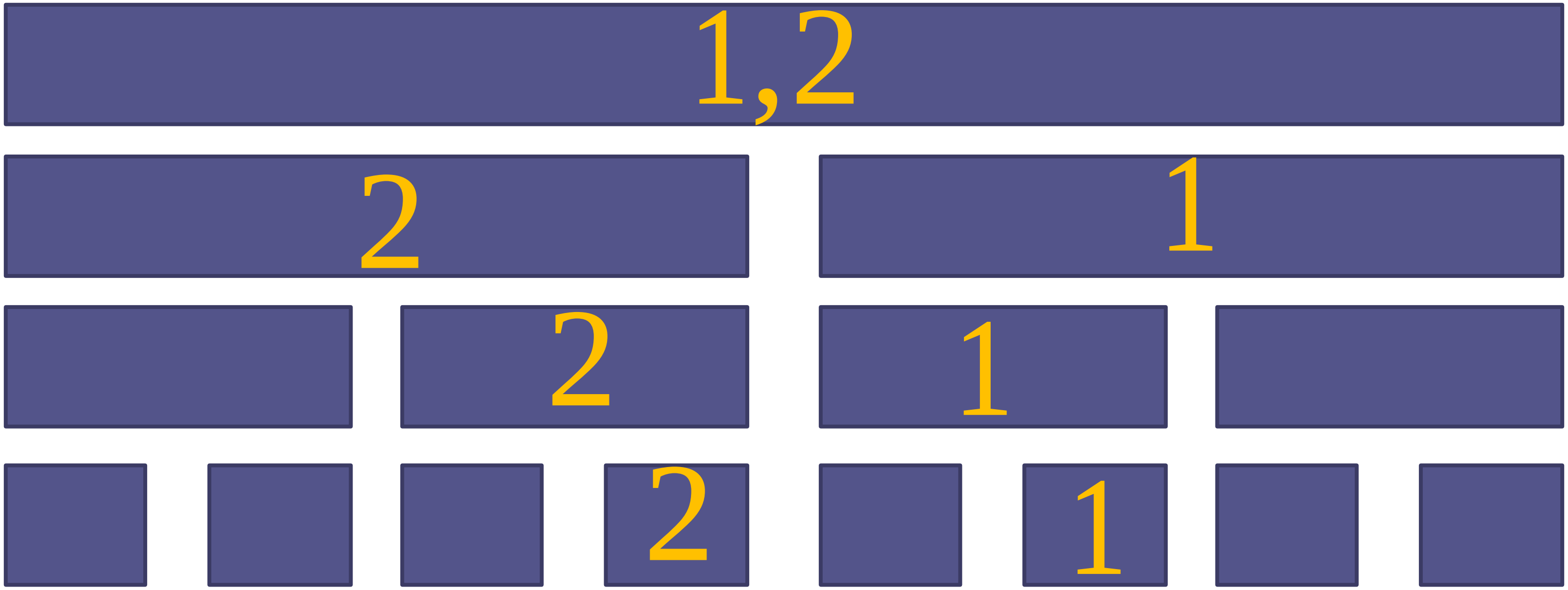
Segment tree

- 区間[L,R]に対し、Rを含む区間の配列にLを昇順に入れておく
  - 区間[1,5]を追加する場合(0-origin)



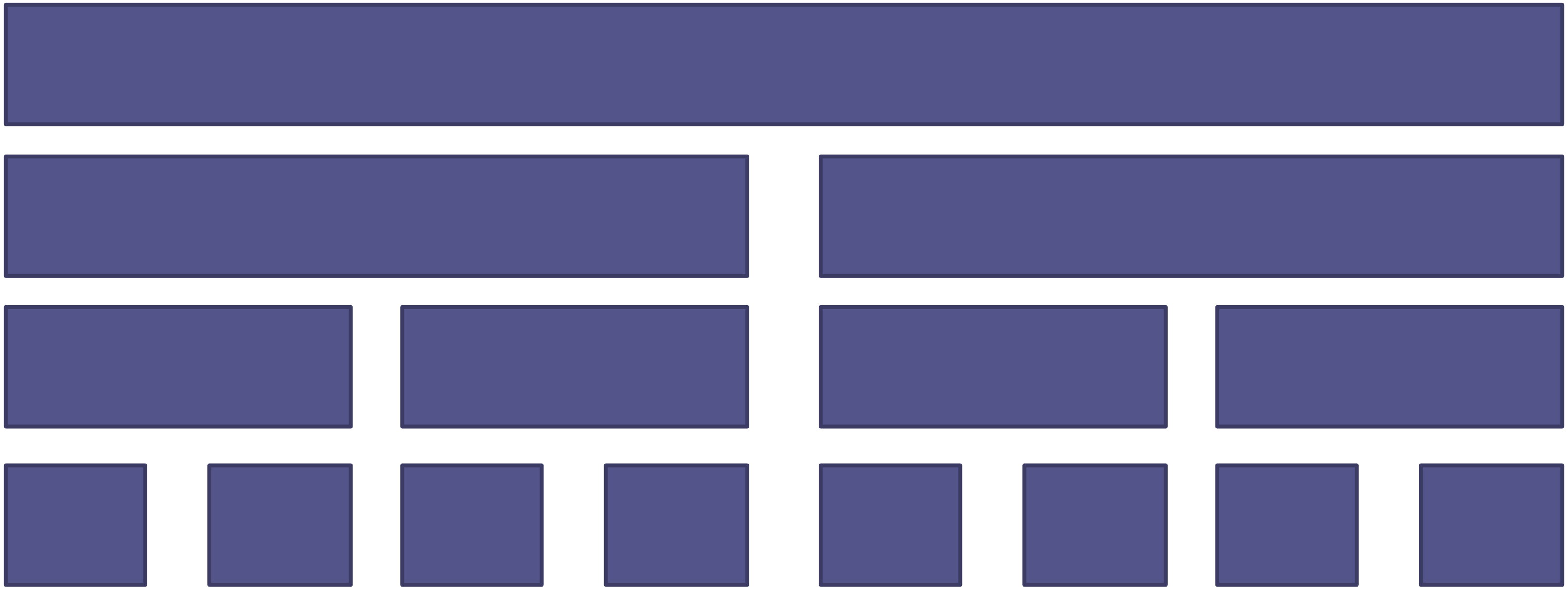
Segment tree

- 区間[L,R]に対し、Rを含む区間の配列にLを昇順に入れておく
  - 区間[2,3]を追加する場合(0-origin)



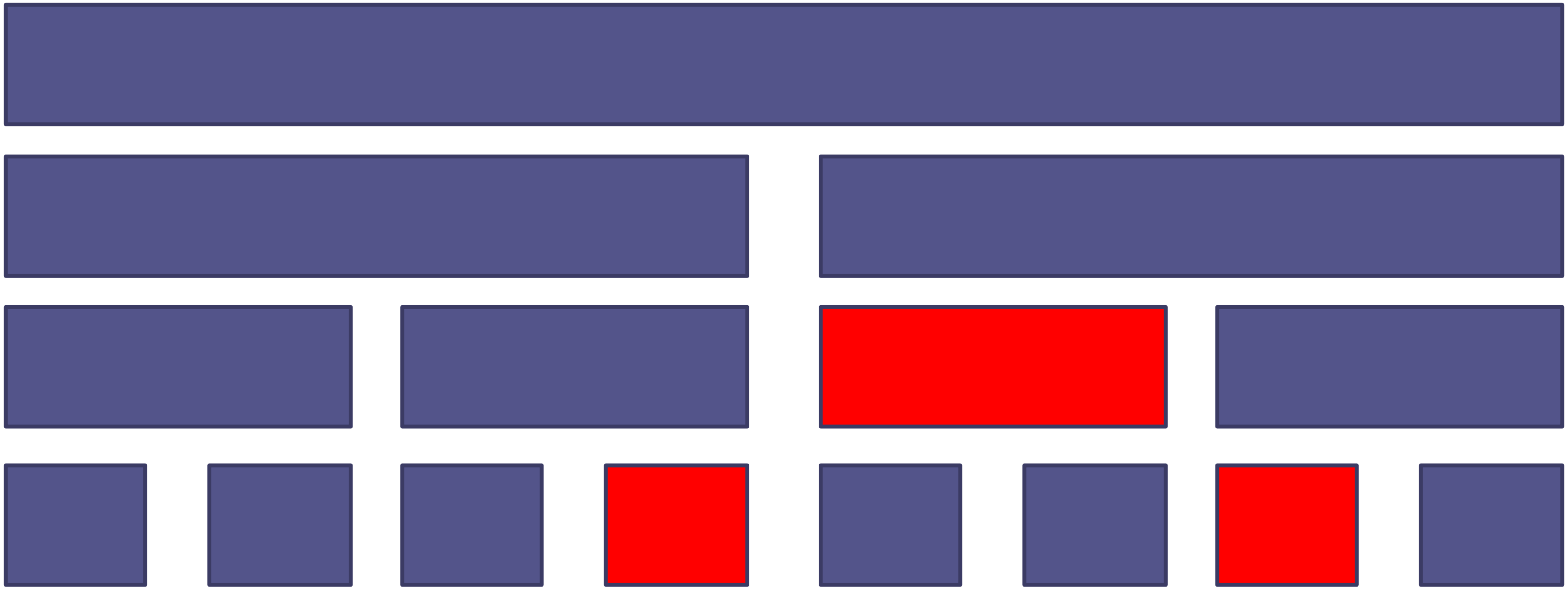
Segment tree

- 区間[3,6]に右端が含まれ、左端が3より前にある区間は?



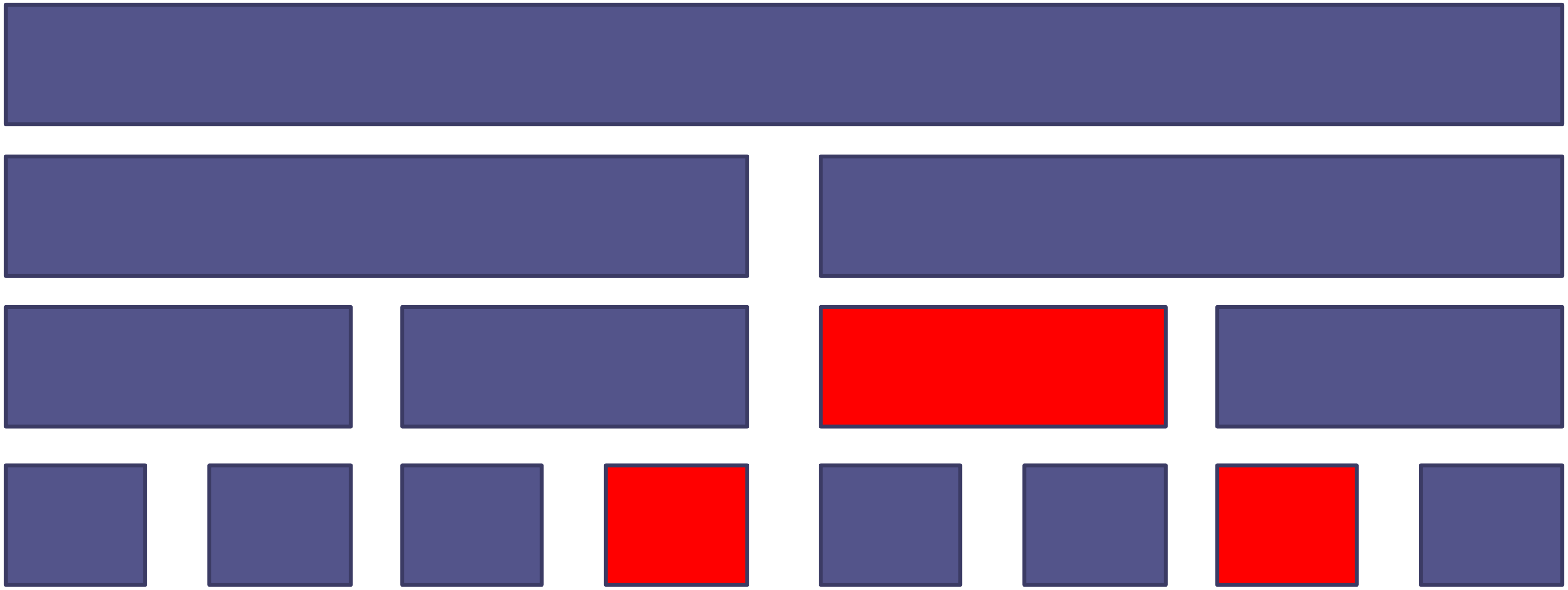
Segment tree

- 区間[3,6]に右端が含まれ、左端が3より前にある区間は?
  - ここに書かれている値のうち3未満のものに対応



Segment tree

- このようなノードで、値がある値以下のものを全部見れば、隣接点がわかる
  - 一度見たら消しておけば余計に見なくて済む



満点解法①

- [L,R]間で始まってRより後で終わる区間についても同様にsegment treeを作る
- segment treeで隣接する頂点の列挙はできるので、この上で幅優先探索等を行えば連結成分は列挙できる

満点解法①

- あとは、二部グラフ判定ができればよい
- これは、(各頂点が二部グラフのどちら側にあるかはわかるので)問題文の通り、実際にスタックに積んだりスタックから取り出したりして判定してやればいい

満点解法①

- 計算量は、
- 配列に値を追加するときは、追加を昇順に行っていけばO(Nlog N)
- 削除も全部でO(Nlog N)
- グラフの探索で見る辺の本数もO(N)
- 全部でO(Nlog N)でできた
- 定数倍が重いので、vectorを使わないなどの工夫をしないと通らないかもしれません

満点解法②

満点解法②

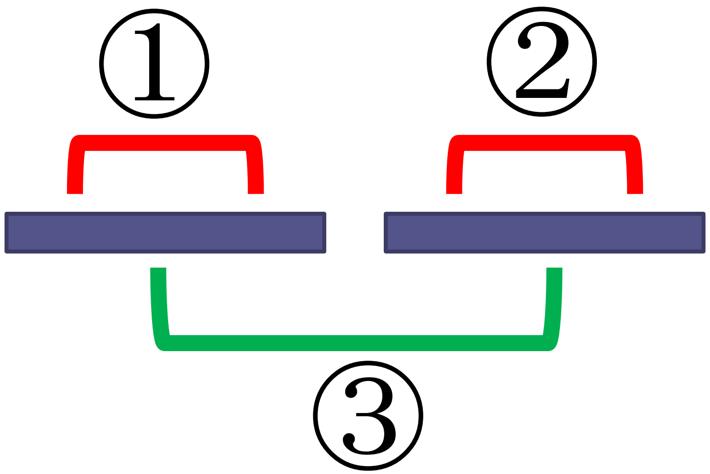
- 分割統治

満点解法②

- 分割統治法
- 区間を半分にして再帰していく
- [L,R]に含まれる区間を以下の3種類に分ける
- M=(L+R)/2として、
  - ①[L,M]に完全に含まれる区間
  - ②[M+1,R]に完全に含まれる区間
  - ③[M,M+1]をまたぐ区間

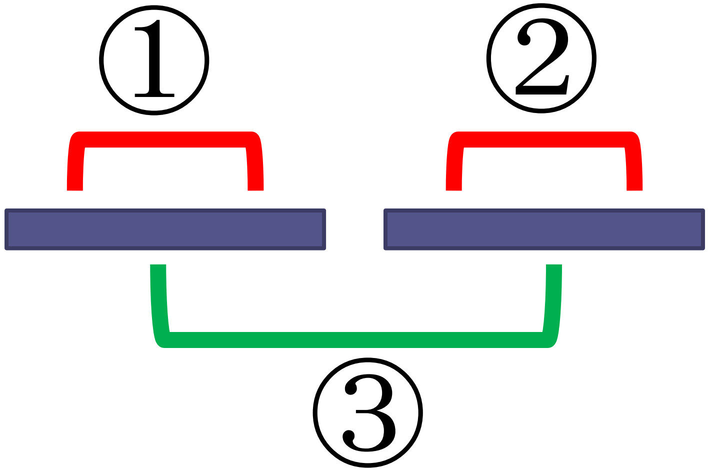
満点解法②

- 上の2つに関しては、再帰してその中で見る
- 残りの3つについて辺を張ればよい
- 3つめと4つめは対称なので、3,5個目のみ考える
  - ①の区間と①の区間の間の辺
  - ②の区間と②の区間の間の辺
  - ①の区間と③の区間の間の辺
  - ②の区間と③の区間の間の辺
  - ③の区間と③の区間の間の辺



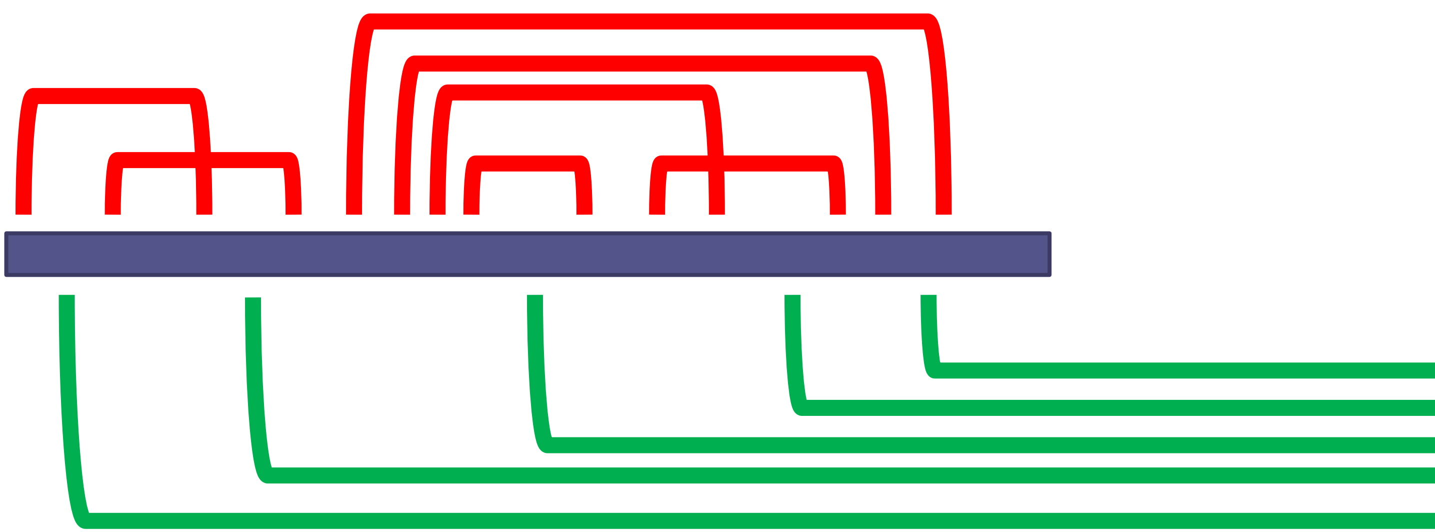
満点解法②

- ①の区間と③の区間の間の辺
- ③の区間は、左端のみ関係ある
- ③の区間を左端の座標の昇順にソートしておく



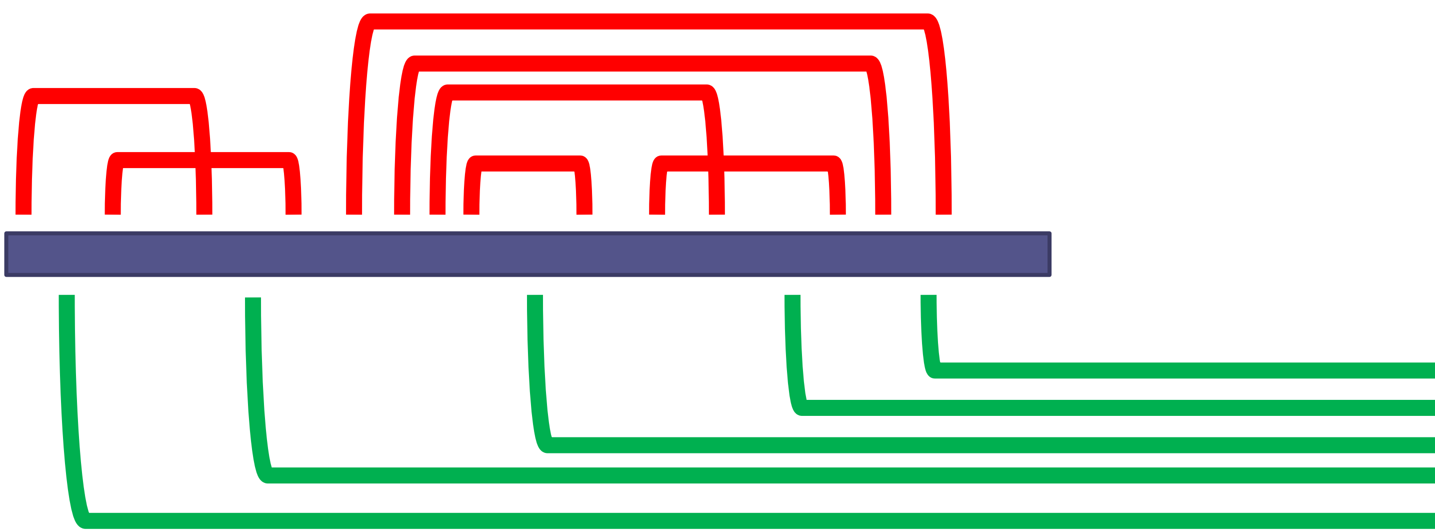
満点解法②

- ①の区間と③の区間の間の辺
- ③の区間は、左端のみ関係ある
- ③の区間を左端の座標の昇順にソートしておく



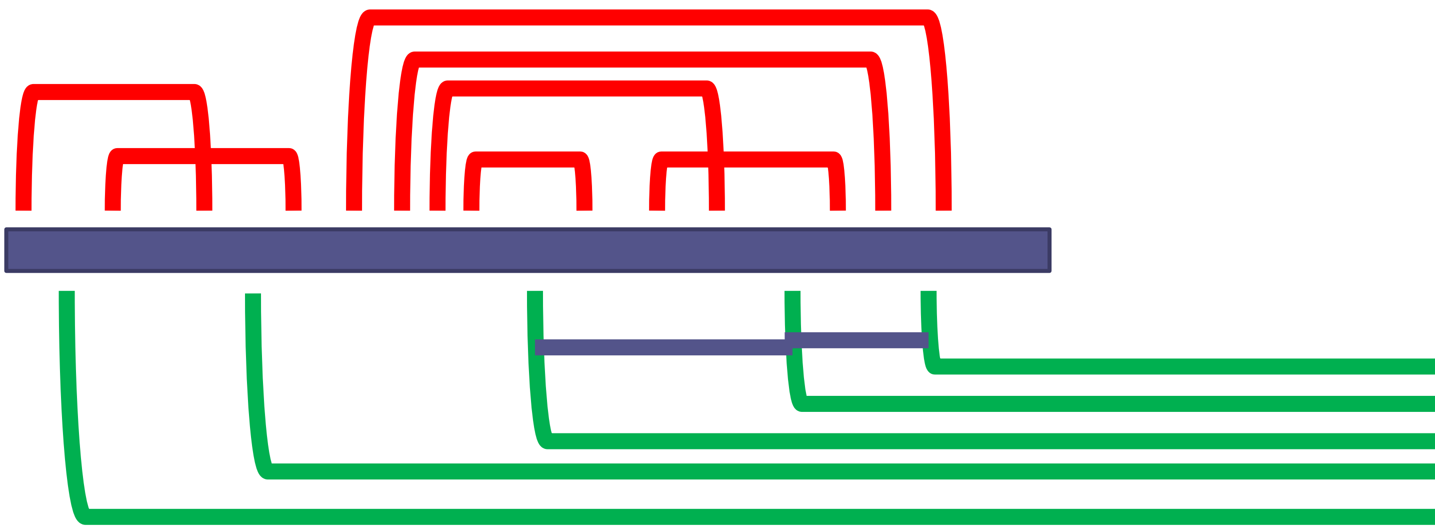
満点解法②

- 各①の区間に対し、それに左端が含まれるような「②の区間の区間」を求めておく
- この区間の区間すべてに辺を張るとまだ張りすぎ



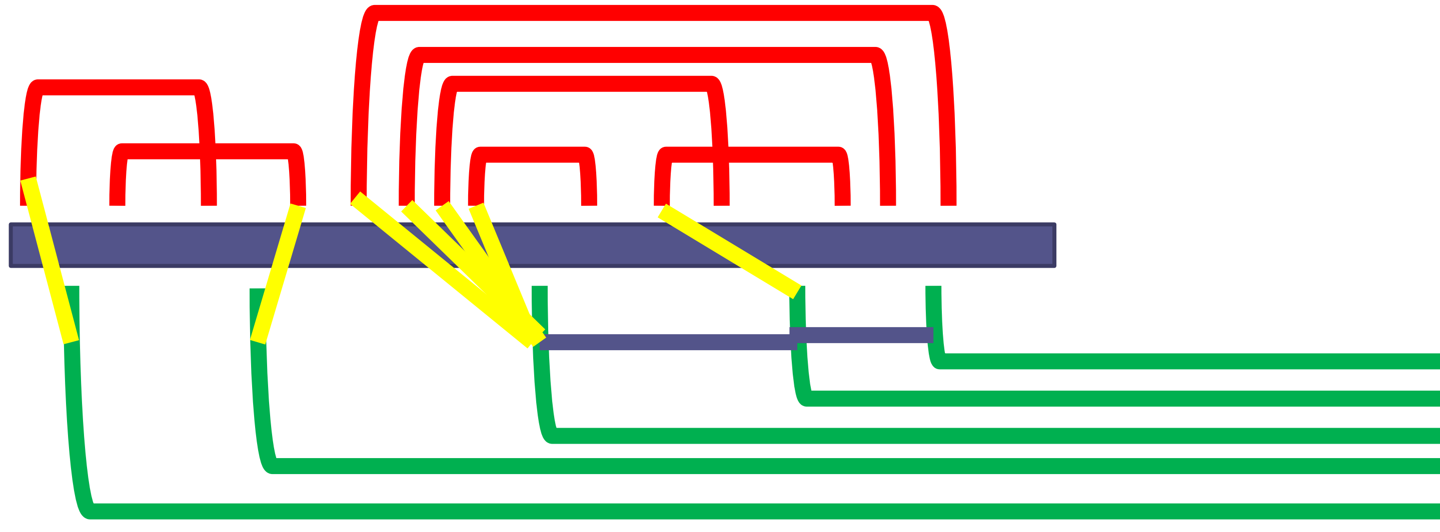
満点解法②

- となりあう2つの③の区間に対し、それを両方含む①の区間が存在するならその間にコスト0の辺を張る



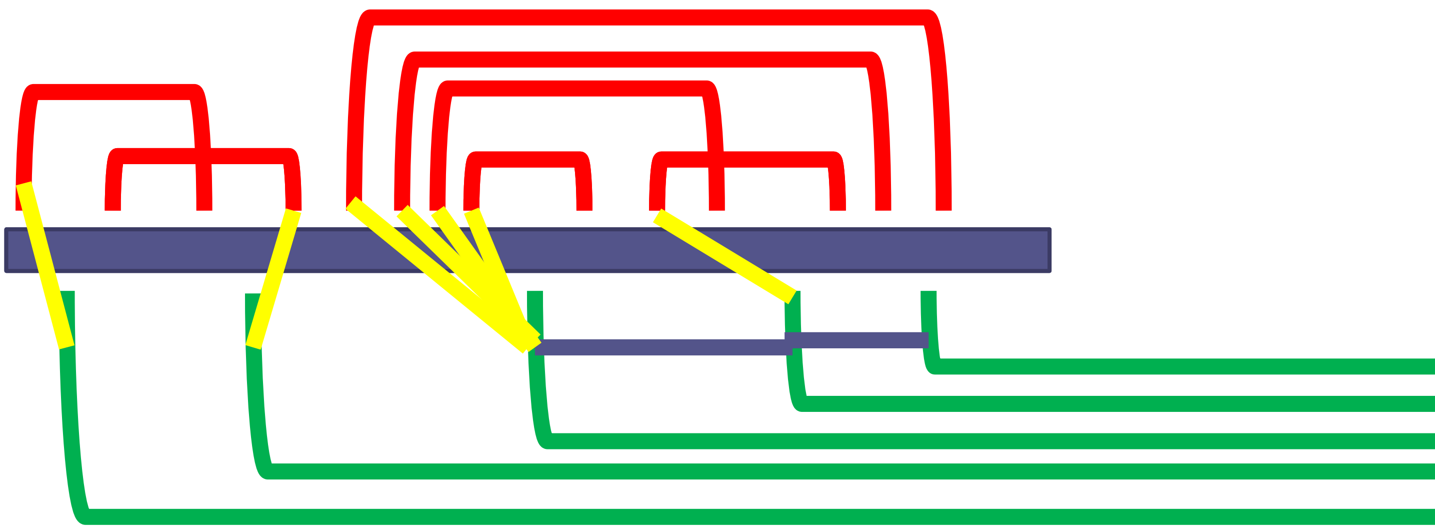
満点解法②

- さらに、各①の区間に対し、それに左端が含まれる③の区間が存在するなら、そのうち一つに辺を張る



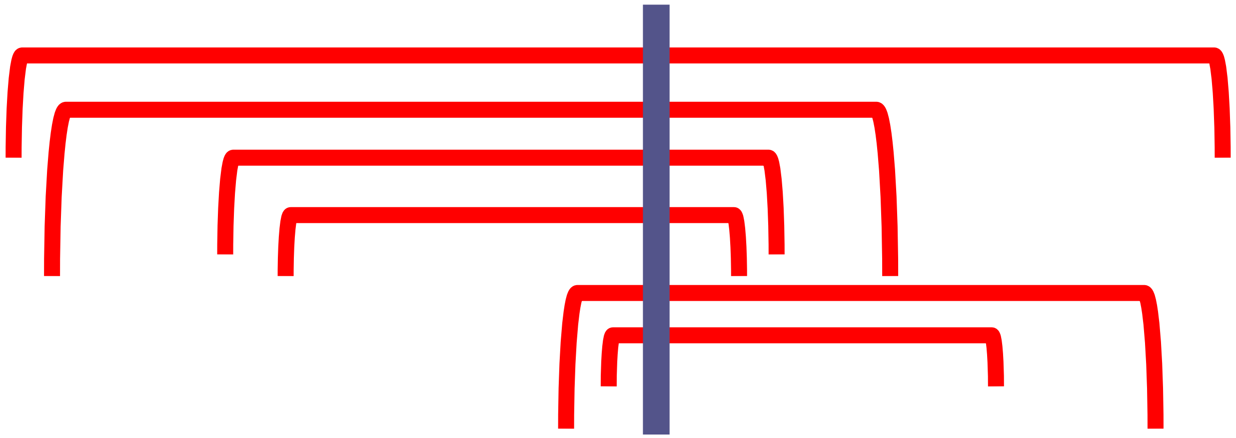
満点解法②

- こうすると、区間の数くらいの辺の数で、二部グラフ性のようなものが等価なグラフが構成できる
- 適切に実装すればO(区間長)でできる



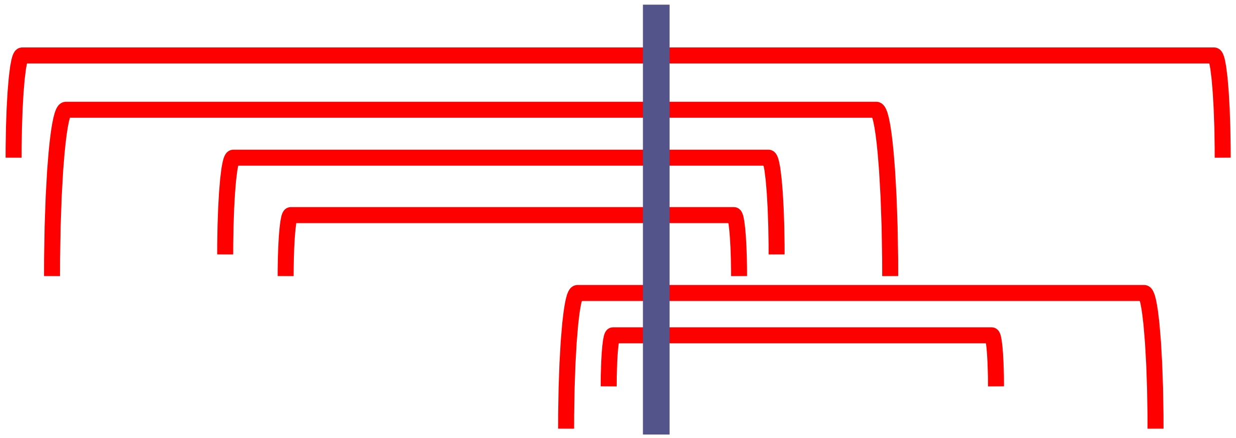
満点解法②

- ③の区間と③の区間の間の辺



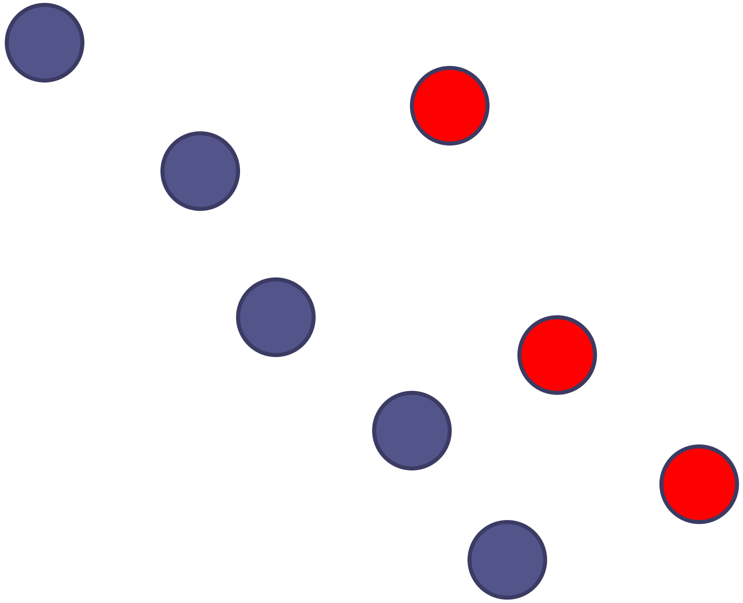
満点解法②

- (始点,終点)のペアを平面上にプロットしておく
- 左下に点のある点と、ない点に分ける



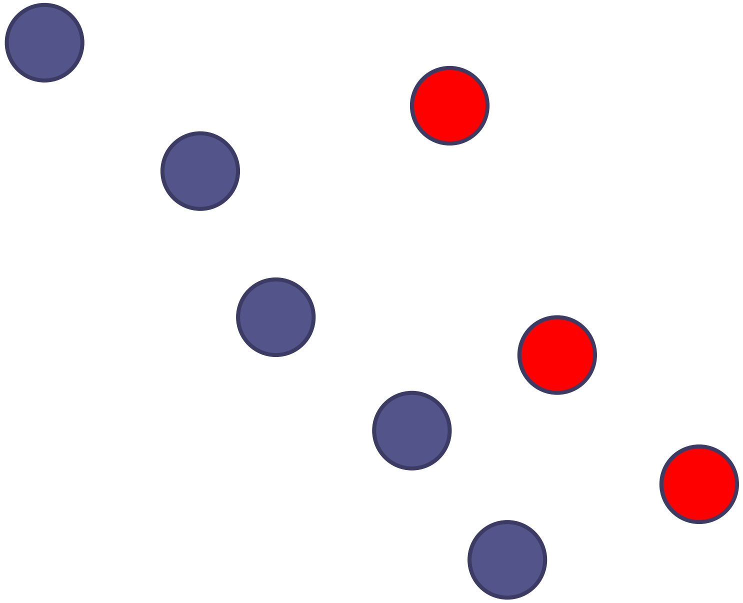
満点解法②

- (始点,終点)のペアを平面上にプロットしておく
- 左下に点のある点と、ない点に分ける
  - 右下-左上の関係にある点対の間に辺を張りたい



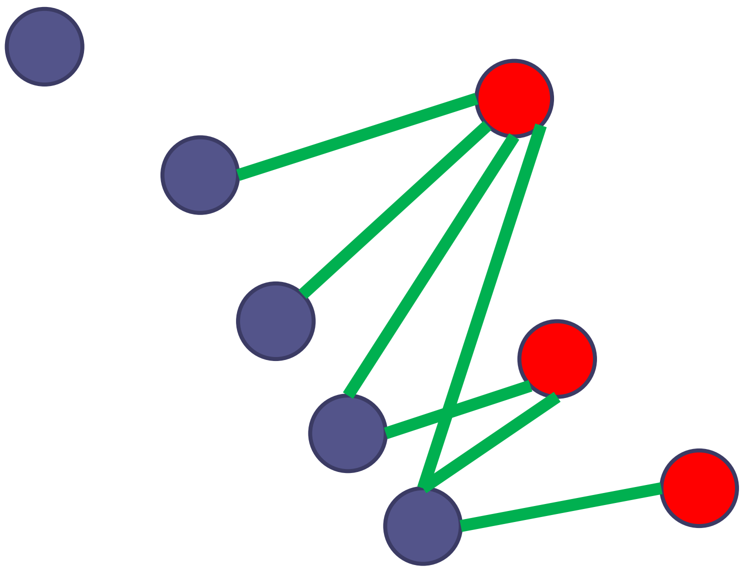
満点解法②

- まず、赤い点に右下-左上の関係にある2点があれば、二部グラフでない
  - 三角形ができるため



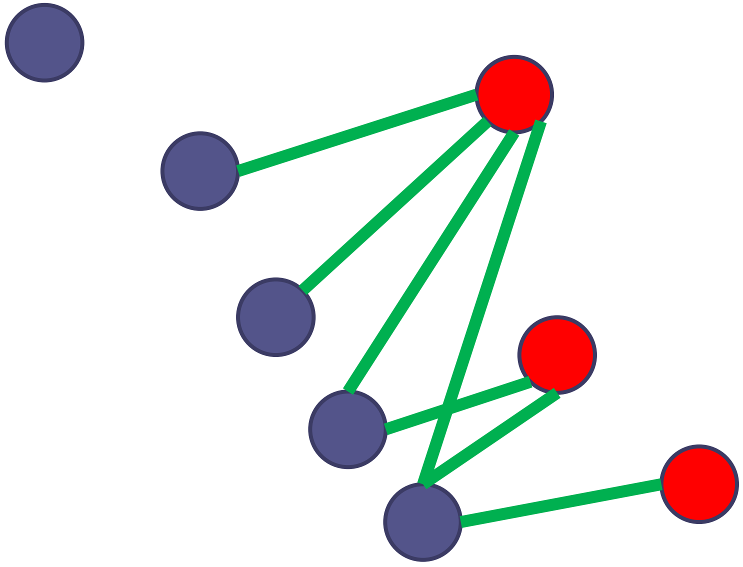
満点解法②

- (始点,終点)のペアを平面上にプロットしておく
- 左下に点のある点と、ない点に分ける
  - 右下-左上の関係にある点対の間に辺を張りたい



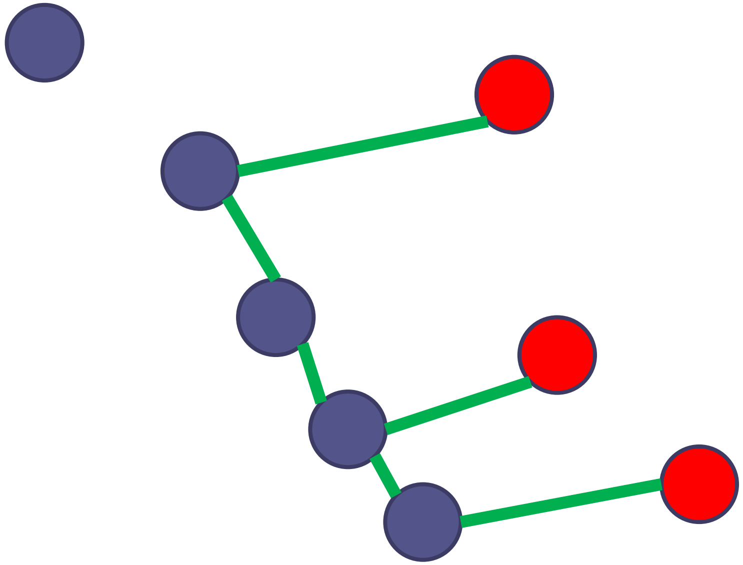
満点解法②

- このままだと辺数はO(N^2)



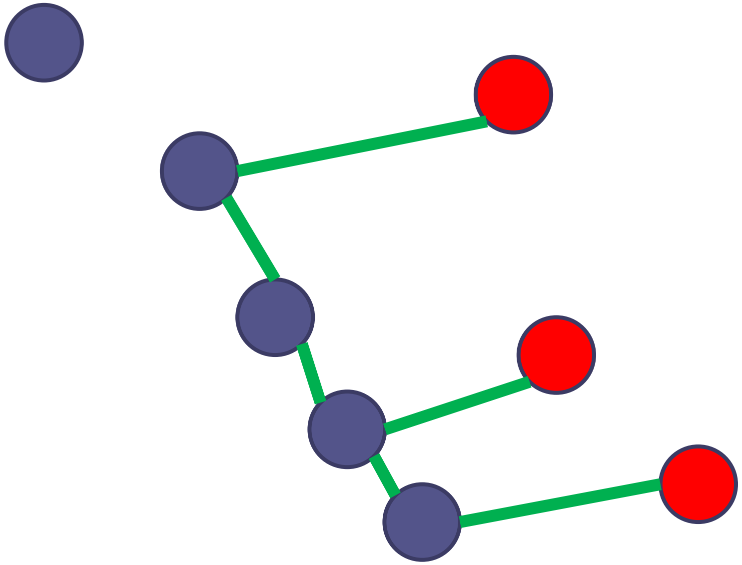
満点解法②

- これも、先ほどと同様に、隣り合う点の間にコスト0の辺を張ることで、O(N)辺で等価なグラフが構成できる



満点解法②

- 適切に実装すればO((点の数)log (点の数))で辺が張れる



満点解法②

- 以上をまとめて、O(Nlog N)でこの問題が解けた

満点解法③〜

- ほかにもいくつか解法があるそうです
- いろいろ見つけてみましょう

得点分布

- 100
- 78 78
- 22 22 22 22 22 22 22 22 22 22 22 22 22 22 22 22
- 10
- 0 0