

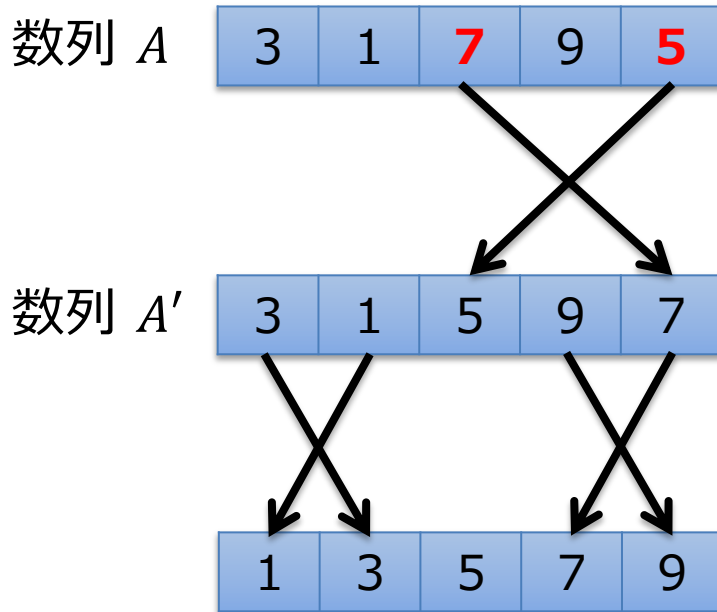
第 12 回 日本情報オリンピック本選 問題5

バブルソート

秋葉 拓哉 / @iwiwi

(東京大学 情報理工学系研究科)

問題



① 数列 A が与えられる

② 数を 2 つ選んで交換する

③ バブルソートする

バブルソートにおける数の交換回数の最小値は？
(②で出来るだけ良い選び方をしたい)

話すこと

1. バブルソートの交換回数とは？
(+10 点解法)
2. 交換回数はどのように変化するか？
(+30 点解法)
3. 満点解法

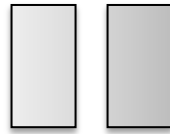
与えられる A に重複がないとして解説します。
重複ありの場合（最後 20 点分）も本質的には同じ。

バブルソートの交換回数とは？

バブルソートの交換回数

数列 A を a_1, a_2, \dots, a_n として,

数列 A をバブルソートした時の交換回数



$i < j, a_i > a_j$ となるような組 (i, j) の個数

大小と左右の関係が逆になっているペアの個数
(ちなみに, **反転数**という名前がついている)

バブルソートの交換回数

$i < j, a_i > a_j$ となるような組 (i, j) の個数

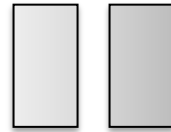
例



3 つ

バブルソートの交換回数

数列 A をバブルソートした時の交換回数



$i < j$, $a_i > a_j$ となるような組 (i, j) の個数

何故？ (略証)

- バブルソートの 1 回の交換で, 反転数は 1 減少
- 完了時すなわちソート済みの配列の反転数は 0

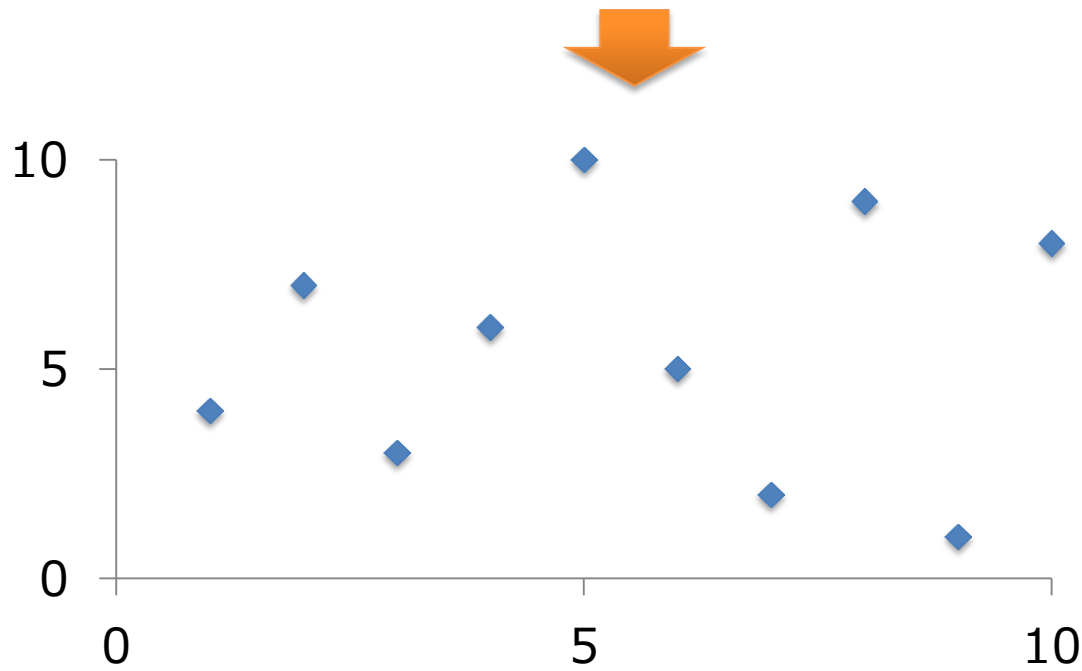
$O(n^3)$ 解法 (10 点)

- 交換する 2 つの数を全通り試す
- 交換回数がどうなるか毎回 $O(n)$ で調べる
(その 2 数に関する部分だけ数えなおす)

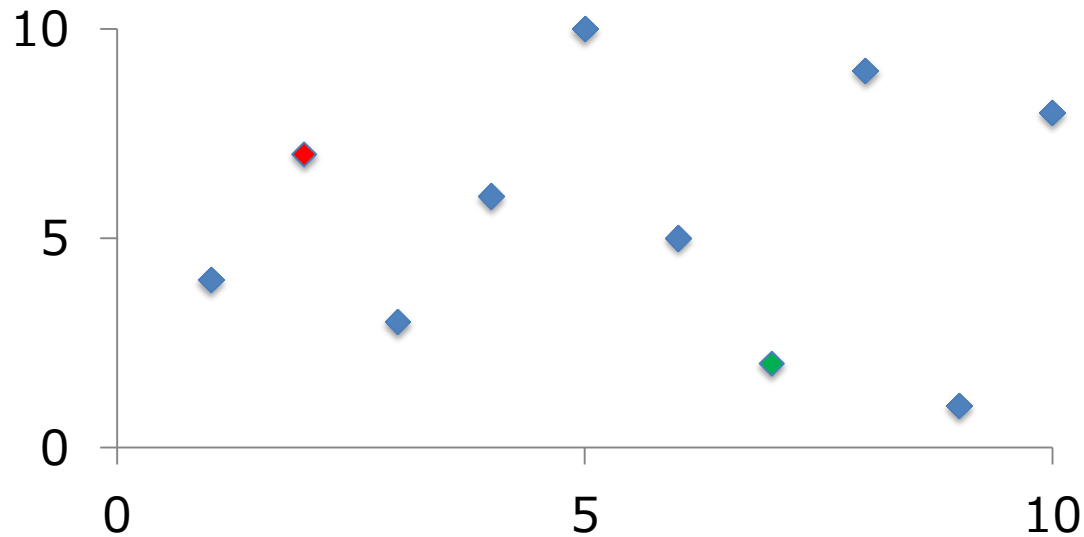
交換回数はどのように変化するか？

交換回数はどうのように変化するか？

数列を平面にプロットして考えてみよう

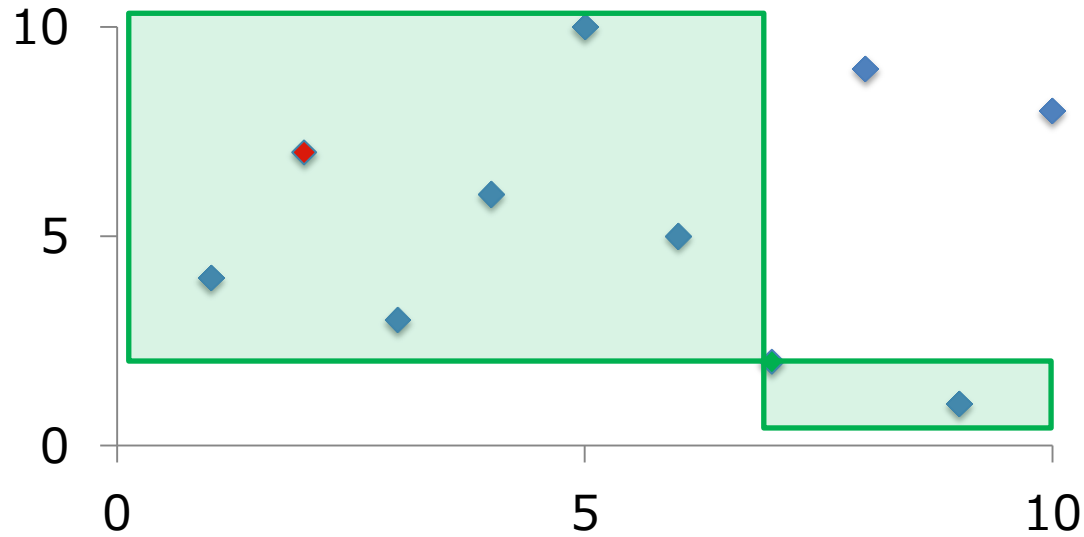


交換回数はどうのように変化するか？



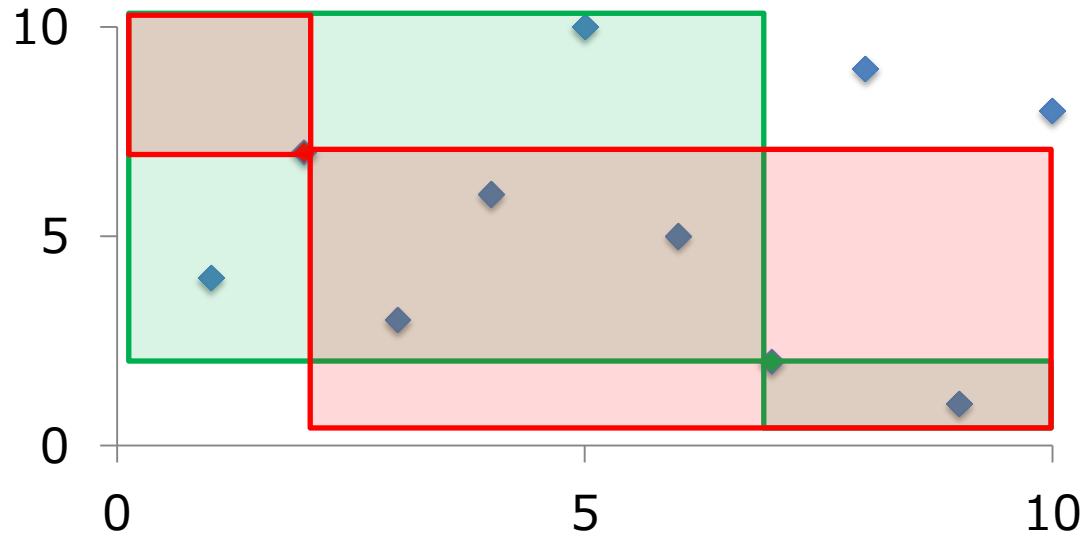
赤と緑を交換した時、
バブルソートの交換回数はいくつ減る？

交換回数はどうのように変化するか？



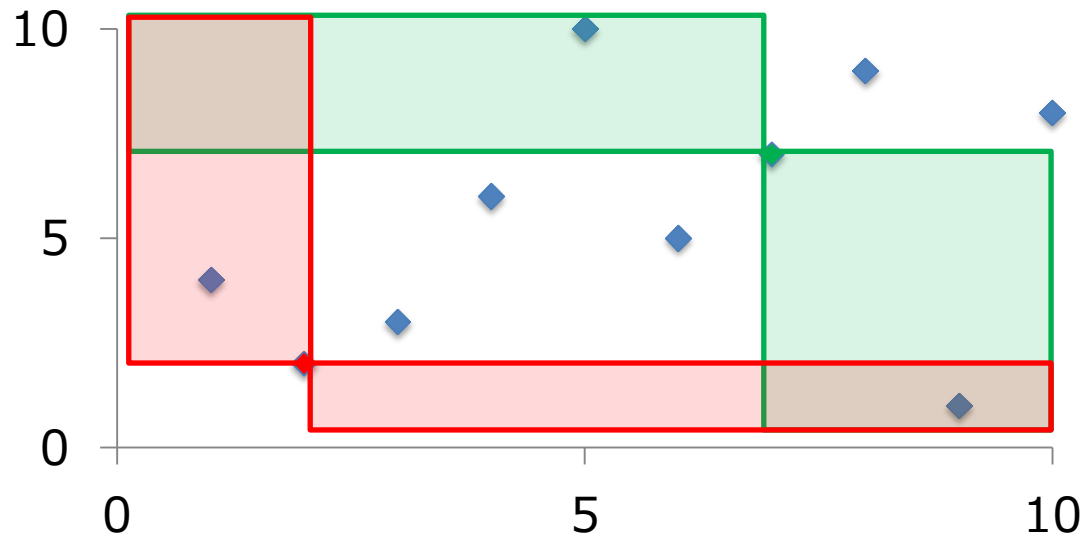
現在，**緑**が反転数に寄与している範囲

交換回数はどのように変化するか？



現在，赤と緑が反転数に寄与している範囲

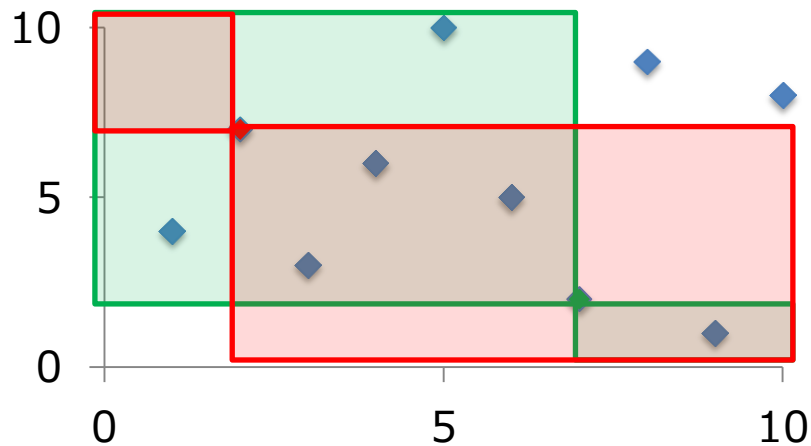
交換回数はどのように変化するか？



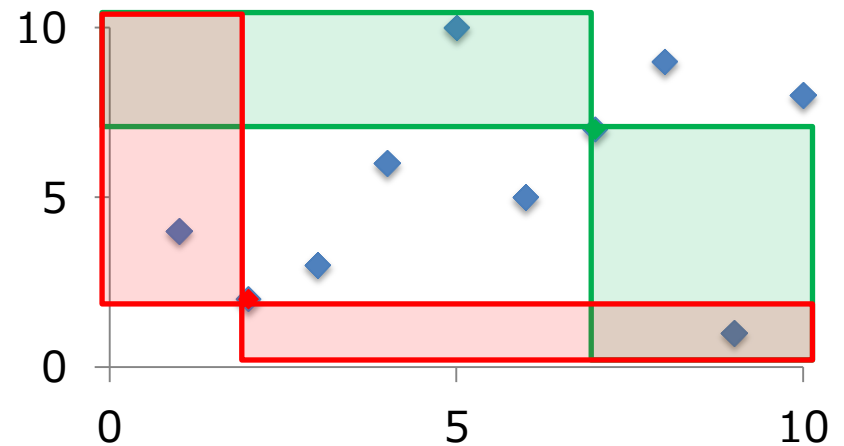
交換後

交換回数はどのように変化するか？

交換前



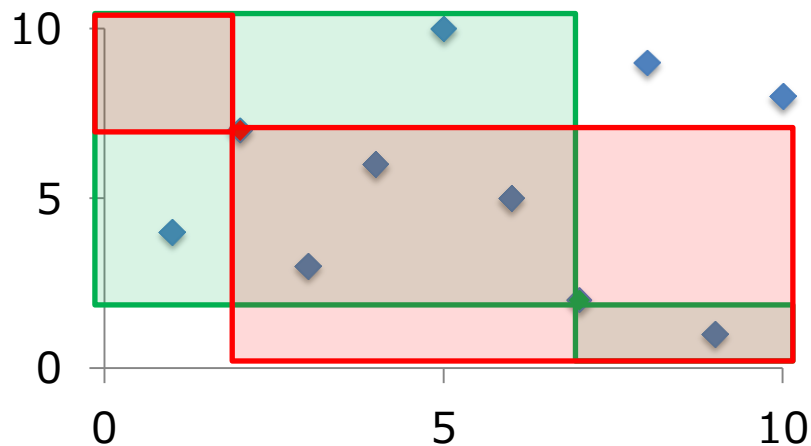
交換後



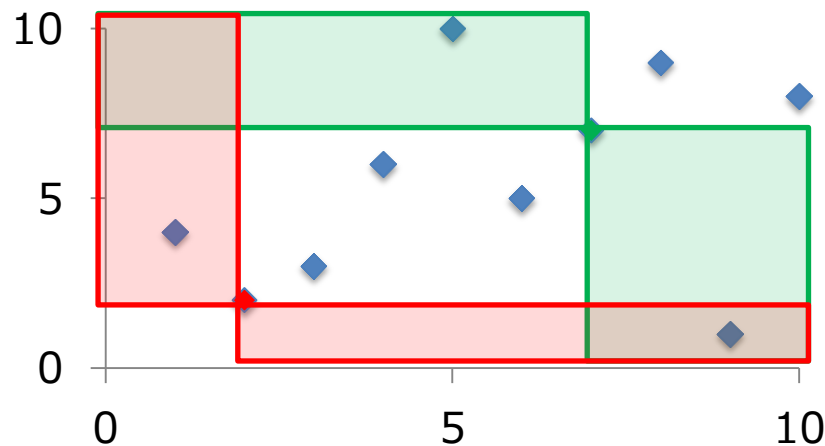
真ん中の長方形がすっぽり消える！
それ以外の部分は変わらない！

交換回数はどうのように変化するか？

交換前



交換後



交換回数の減少：

$$(\text{長方形に含まれる点の数}) \times 2 + 1$$

- $\times 2$: 交換前は二重になってるので
- $+1$: 赤と緑それら自体の分

交換回数はどうのように変化するか？

つまり,

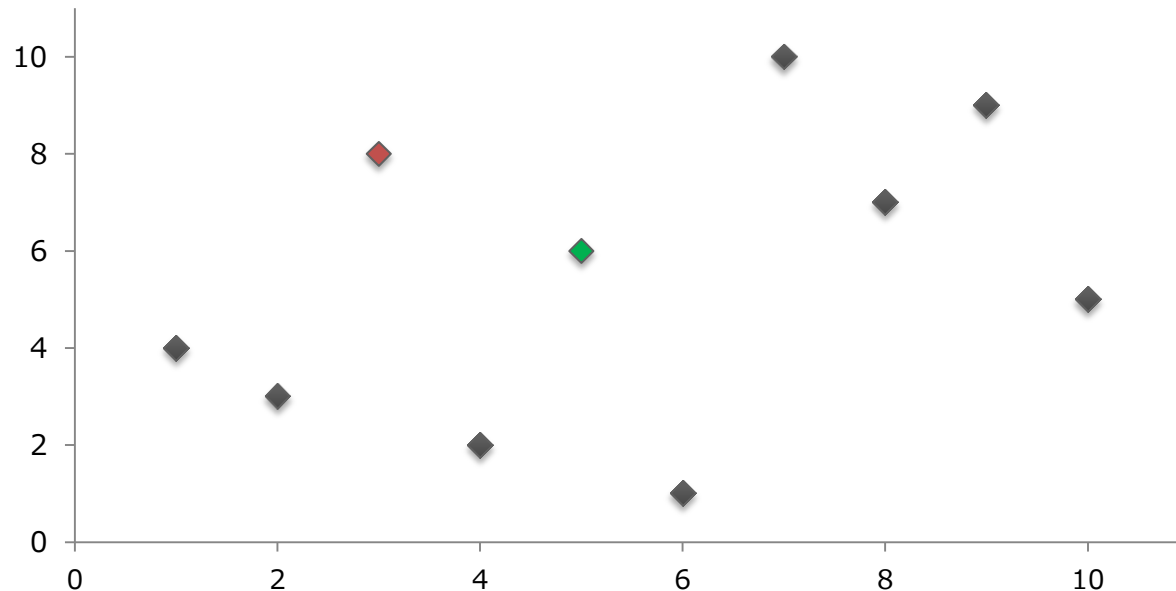
**それを左上・右下にして長方形を描いたら
できるだけ多く点を含むような 2 点を探す
という問題を考えれば良いことになった！**

$O(n^2)$ 解法 (30 点)

- 交換する 2 つの数を全通り試す
- 交換回数がどうなるか毎回 $O(1)$ で調べる
(累積和を前処理として計算しておけばよい)

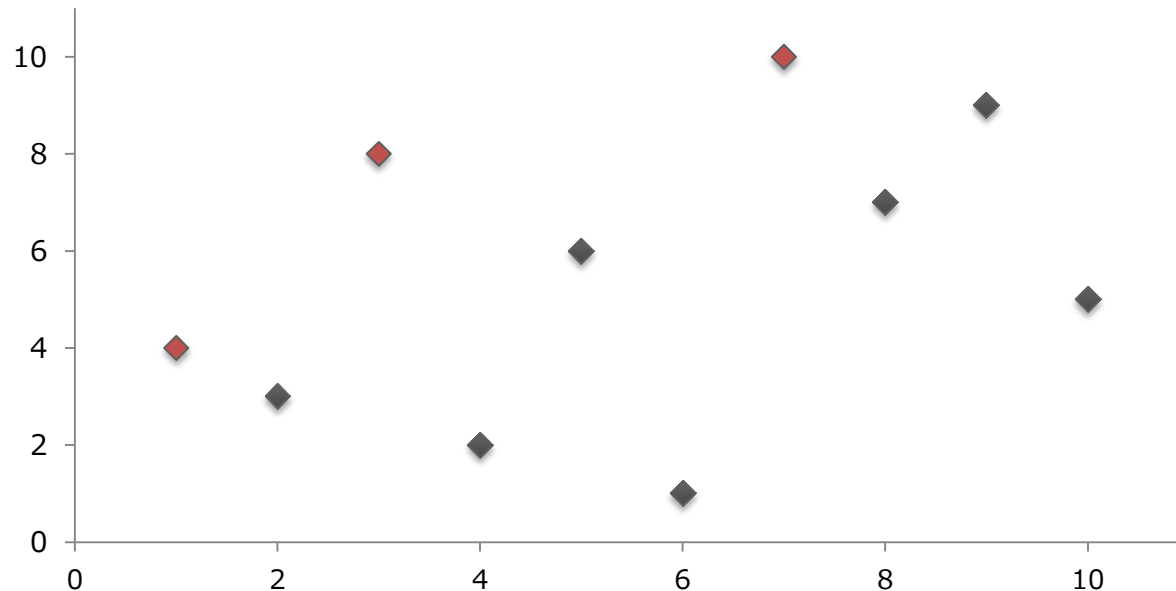
満点解法

考察



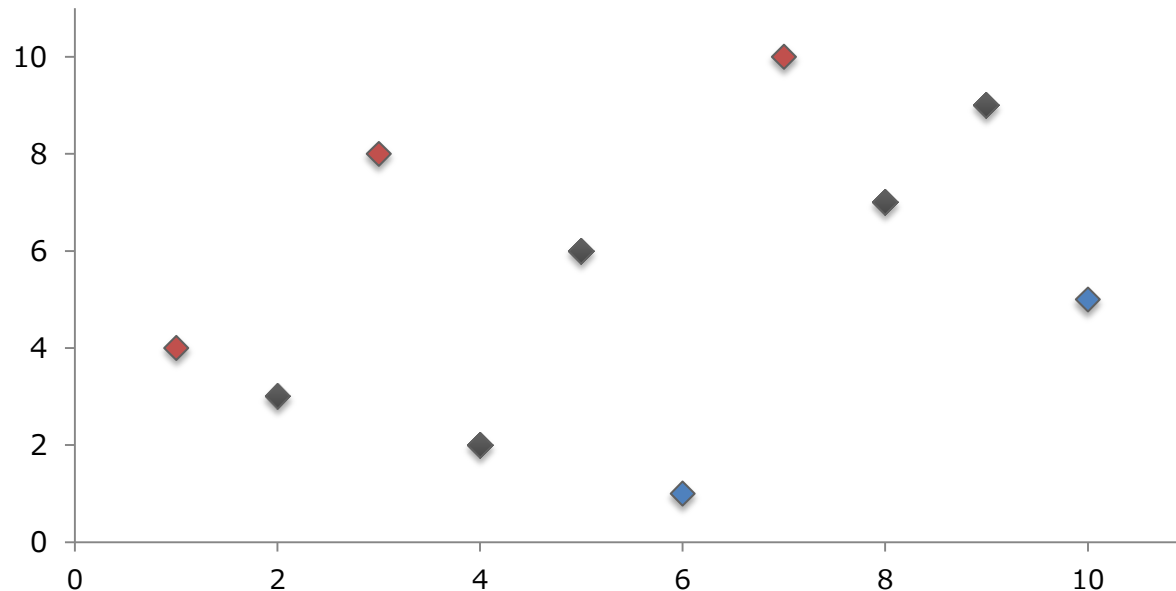
緑を長方形の左上とすることはあり得ない
(緑にするなら赤にしたほうが必ず得)

考察



左上の候補は，自分より左上に点がない点
(上に赤色で示した 3 つだけ)

考察



同様に，右下の候補も絞れる（青い点）
点が 3 種類に分類された！

満点を取りたい……！

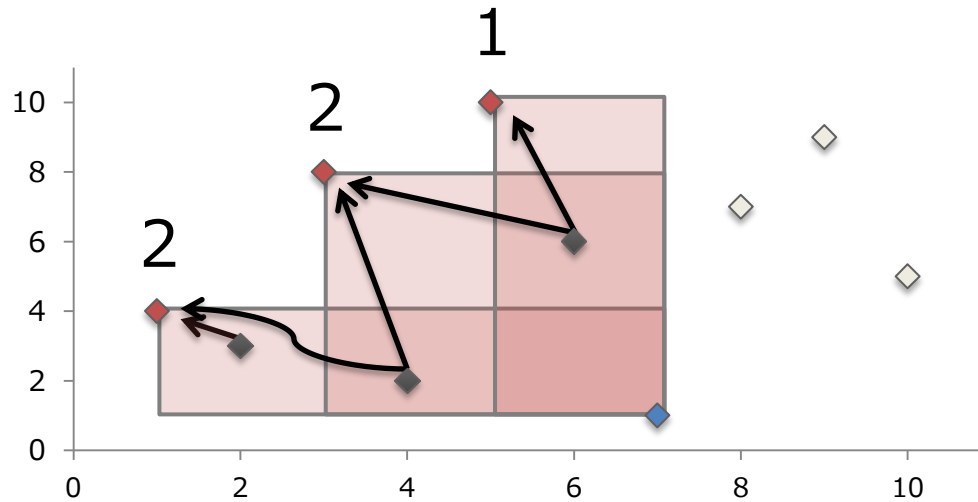
$n = 10^5$ を解くには $O(n^2)$ は無理

- つまり, 交換を全通り試せない……
- うまく一度に大量の可能性を処理しなければならないはず……

どうする？

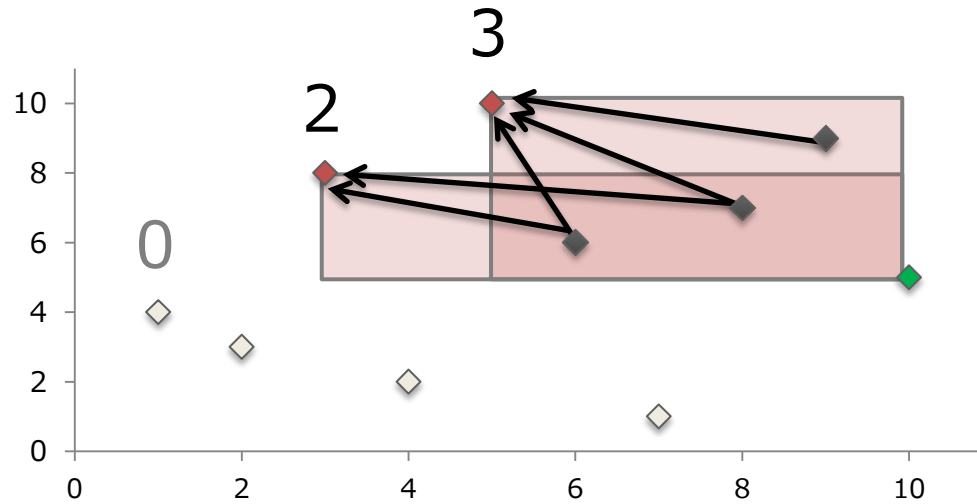
- 右下候補はさすがに全通り試すとする
- それぞれに対して, 最良の左上候補を瞬時に知ることが出来れば良い……出来るか！？

解法を考える



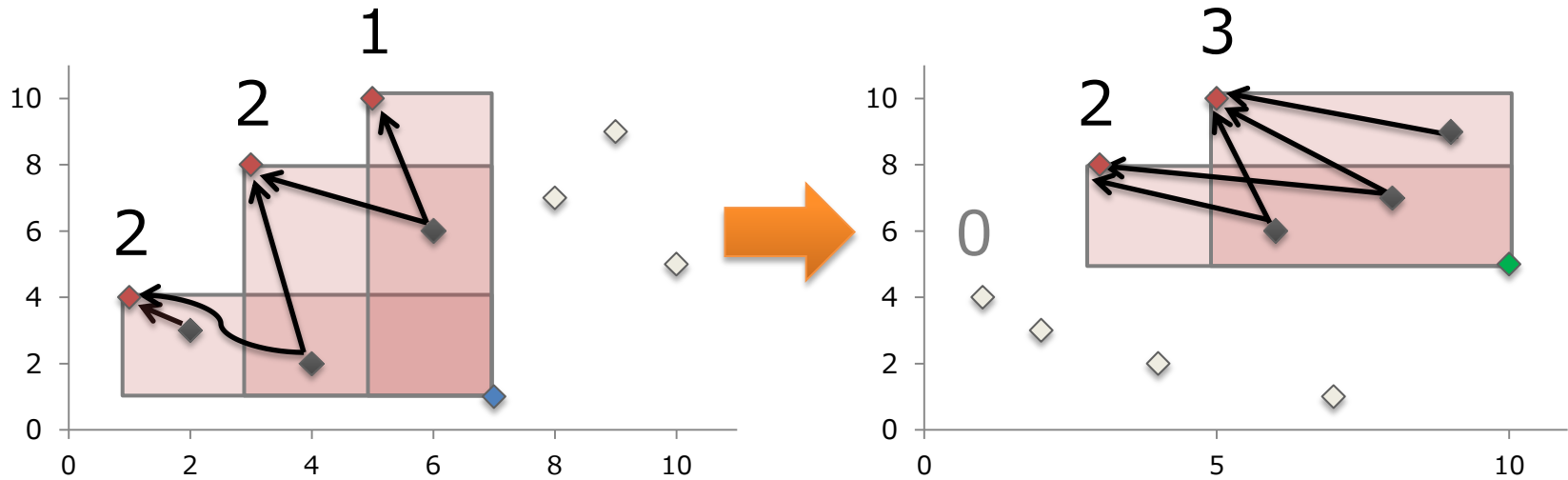
- 青の点を右下にするとしたときの図
- 赤の3点は, スコア 2, 2, 1
(スコア = それを左上にした時に囲える点の個数)

解法を考える



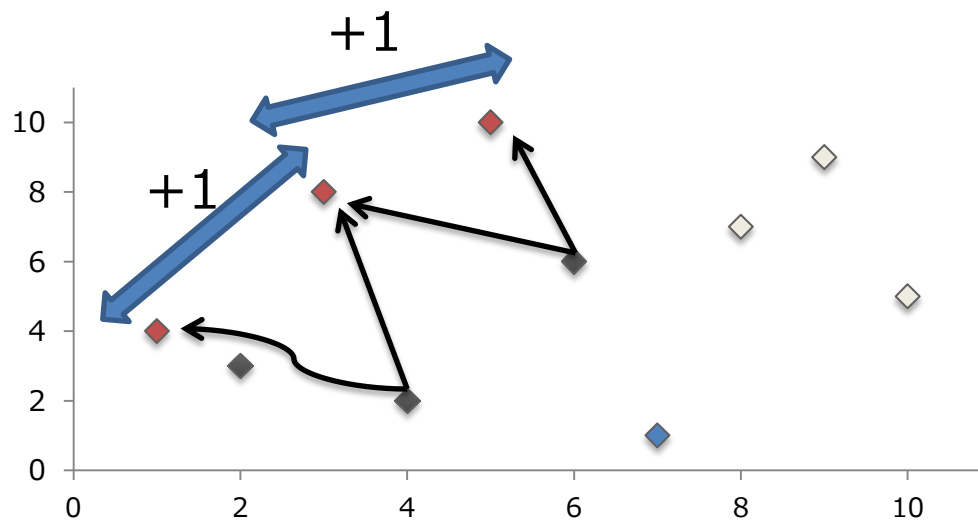
- 緑の点を右下にするとしたときの図
- 赤の3点は, スコア 0, 2, 3

解法を考える



左から右をスムーズに作れないか？

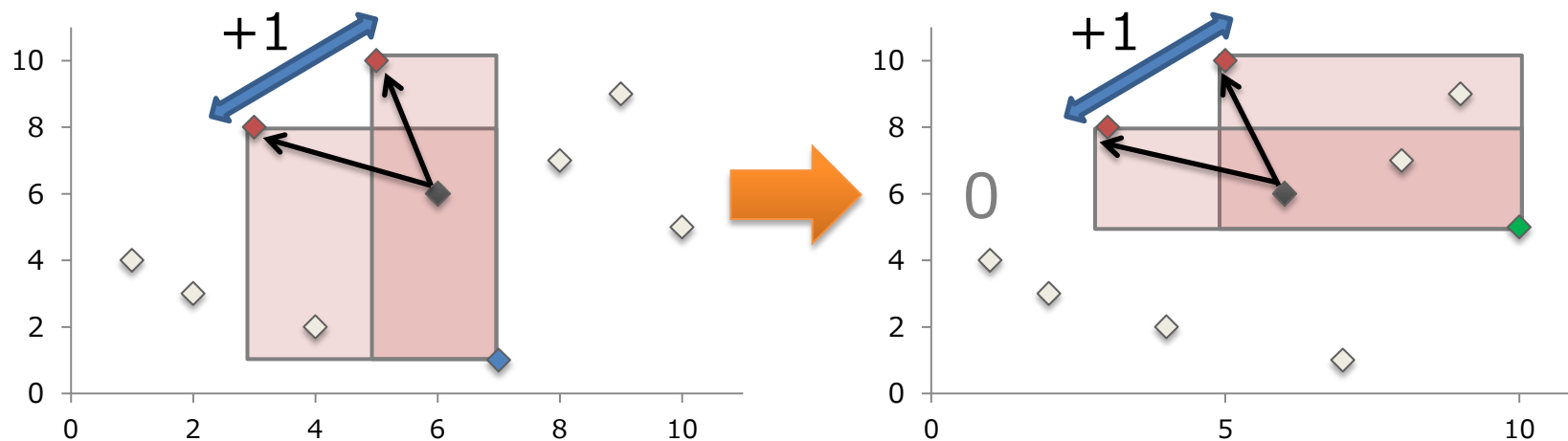
欲しいものはスコアのみ



考察①

赤い点の列に対して、
灰色の点が影響する部分は連続区間
(連続するある部分に +1 している)

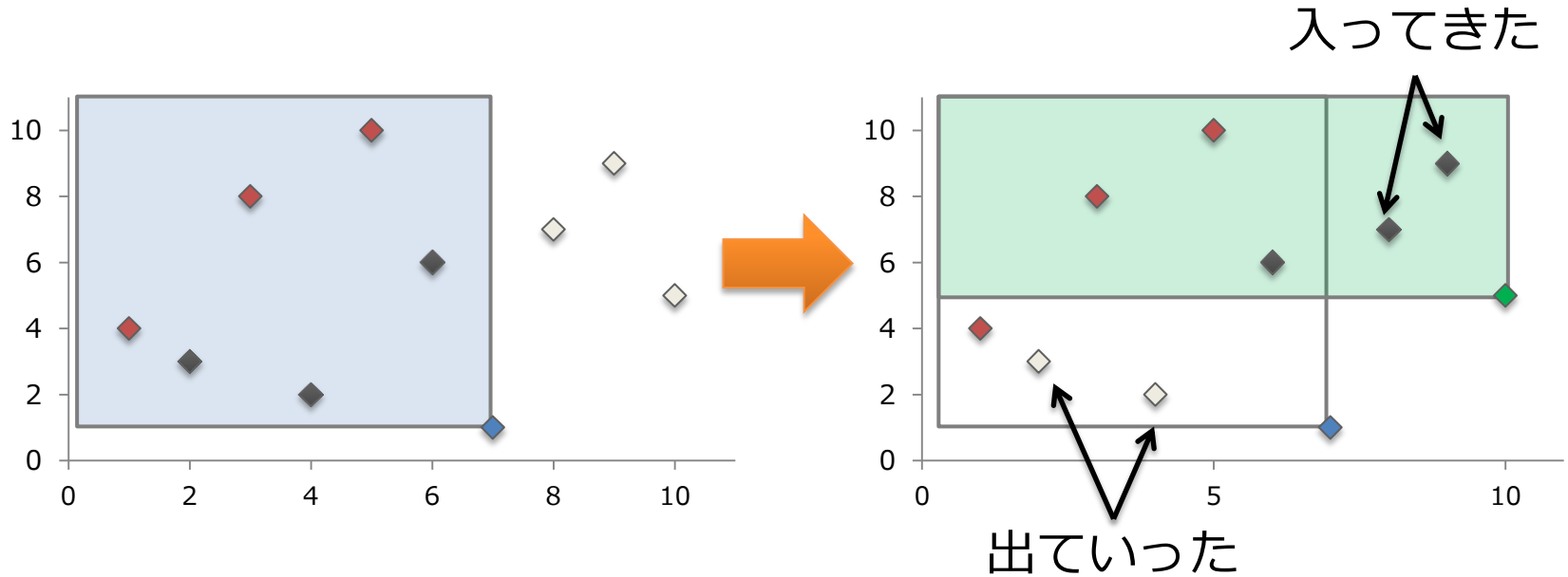
考察



考察②

灰色の点が影響する部分は
右下が違ってても変化しない

解法を考える



考察③

関係する灰色の点の変化は、
「右の点が入ってくる」
「下の点が出ていく」のみ

アルゴリズム

アルゴリズムが作れそうになってきた！

- 左上候補（赤の点）に関するスコアを管理
- 右下候補（青の点）を左から順に試す
 - スコアの最大値を見る
- 他の点（灰色の点）は
 - 青の点の x 座標が自分より大きくなったら「入る」
(赤の点のある区間に +1)
 - 青の点の y 座標が自分より大きくなったら「出る」
(赤の点のある区間に -1)

アルゴリズム

アルゴリズムが作れそうになってきた！

- 左上候補（赤の点）に関するスコアを管理
- 右下候補（青の点）を左から順に試す
 - スコアの**最大値を見る**
- 他の点（灰色の点）は
 - 青の点の x 座標が自分より大きくなったら「入る」
(赤の点の**ある区間に +1**)
 - 青の点の y 座標が自分より大きくなったら「出る」
(赤の点の**ある区間に -1**)

スコアの管理

列において、以下を効率的に行いたい

- 区間に加算
- 最大値の取得

セグメント木を使ってスコアを管理すれば、両方 $O(\log n)$ で実現可能

各点は、1 回入って、1 回出る。

よって、**全体では $O(n \log n)$!**

セグメント木については、以下などを見て下さい。

「プログラミングコンテストでのデータ構造」

(<http://www.slideshare.net/iwiwi/ss-3578491>)



↑ 載ってます

ところで

- 交換回数をどれぐらい減らせるかは分かった
- ところで、最初の列 A のバブルソートの交換回数はどうやって求めるのか？
 - $O(n^2)$ かけて実際にやるのは無理……

2 つの $O(n \log n)$ の方法

- Binary Indexed Tree を使って数える
- 分割統治法により数える



両方載ってます↑

点数分布

