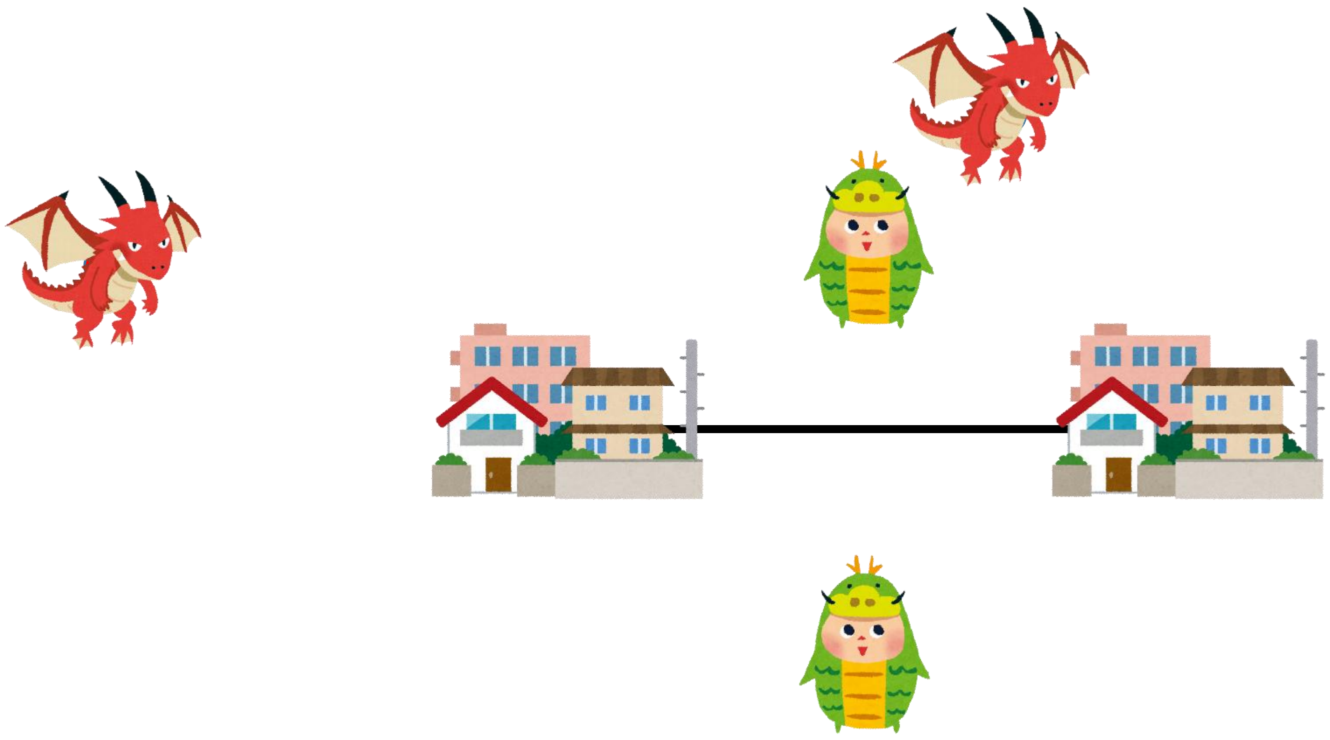
情報オリンピック
２０１７年春合宿
四日目
ドラゴン２

東京大学
理学部数学科四年
笠浦一海

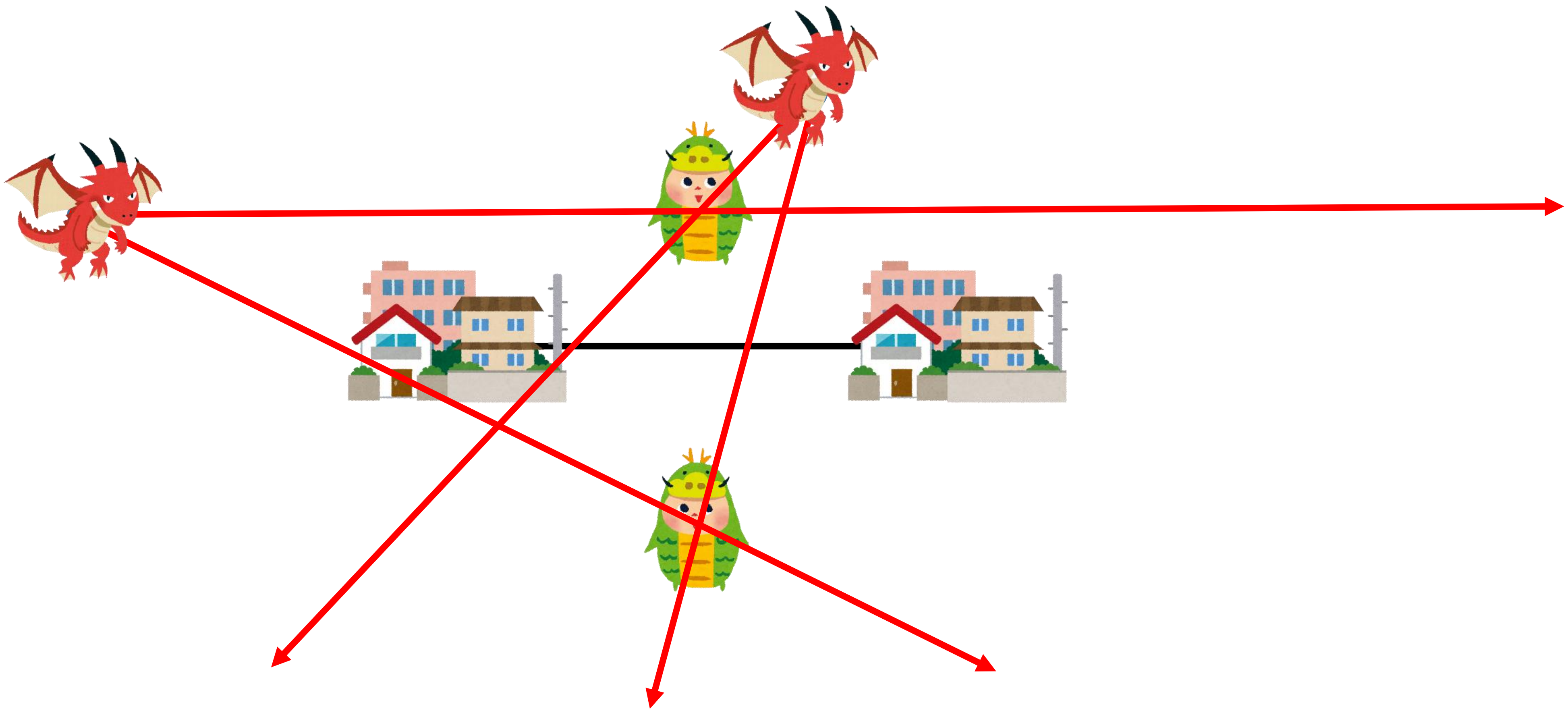
問題概要

- たくさんの点と一つの線分XYが与えられる
- 点たちはいくつかの種類にわけられている
- 次のようなクエリに答えよ
- 「種類aと種類bが与えられるので、種類aの点と種類bの点の組（A, B）で、半直線ABが線分XYと交わるようなものの個数を求めよ」

問題概要



問題概要



Subtask 1

Subtask 1

- N<=3000
- O(N^2)で間に合う
- 点AとBについて、半直線ABが線分XYを交わるかどうかをO(1)で判定できれば良い

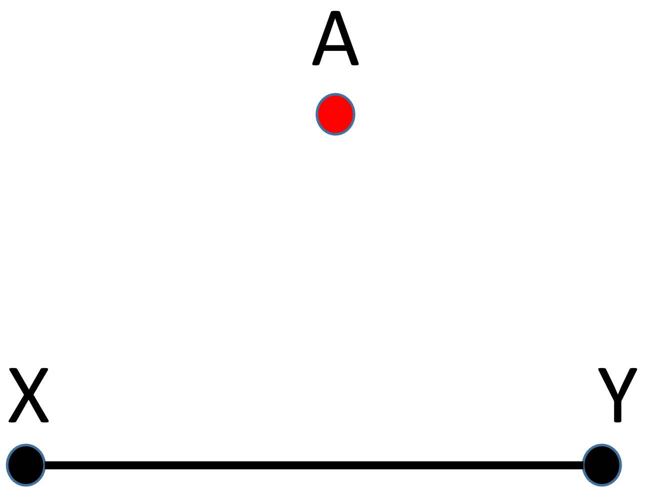
Subtask 1

- 方法1
- 直線AB, 直線XYの交点を計算して, 半直線AB, 線分XY上にあるか判定する
- もっとマシな方法はないか？

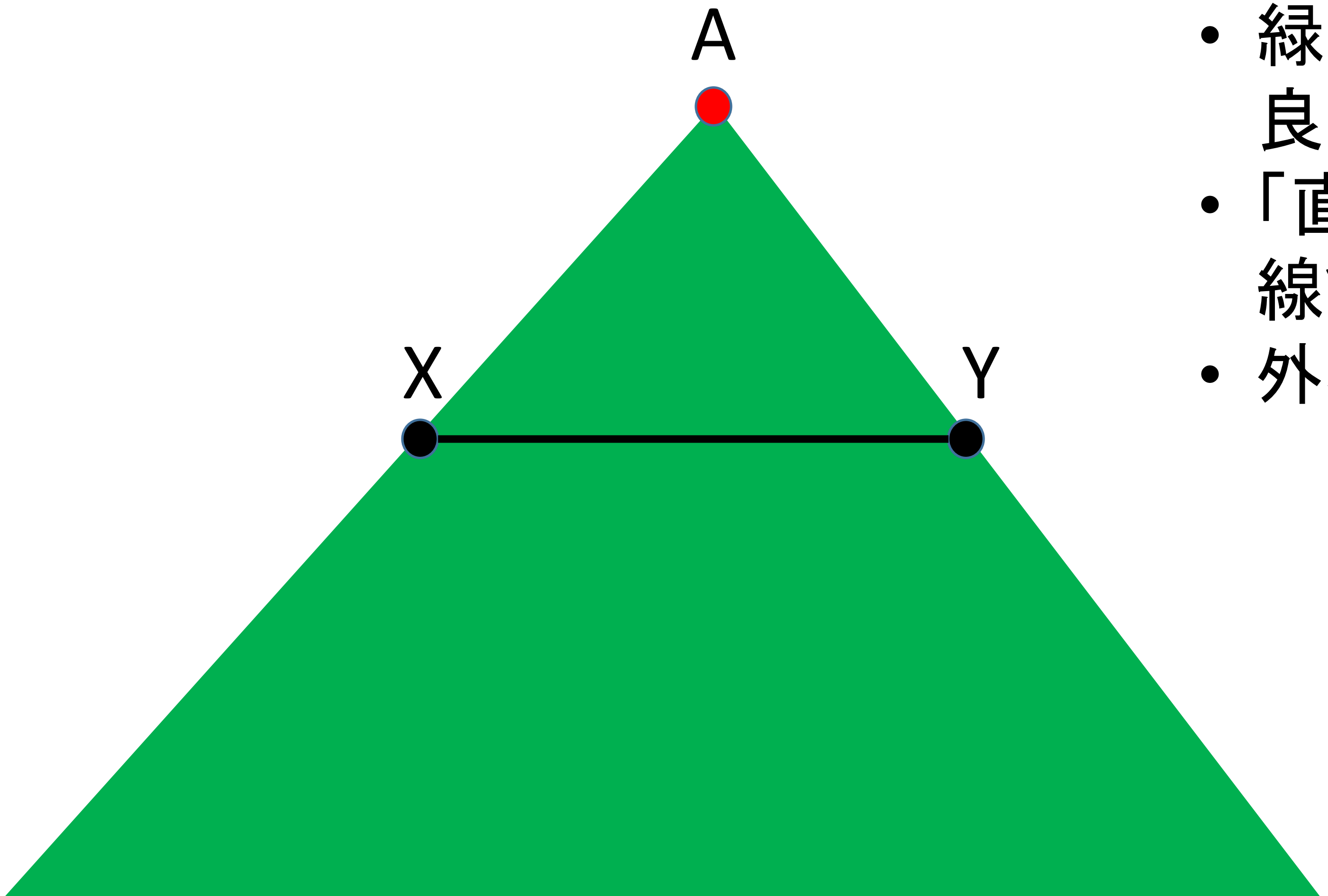
Subtask 1

- 方法２
- Aをfix
- Bがどの範囲にあるときに条件が満たされるか？

Subtask 1



Subtask 1



- 緑の三角形の領域にBが存在すれば良い
- 「直線XAについてYとBが同じ側＆直線YAについてXとBが同じ側」ならよい
- 外積を使って計算できる

Subtask 1

- これでSubtask 1が解けた
- すべての(A, B)のペアについて条件を満たすか計算
- 時間O(N^2+Q)

Subtask 1

- これでSubtask 1が解けた
- すべての(A, B)のペアについて条件を満たすか計算
- 時間O(N^2+Q)
- 15点

Subtask 1

- 注意点

Subtask 1

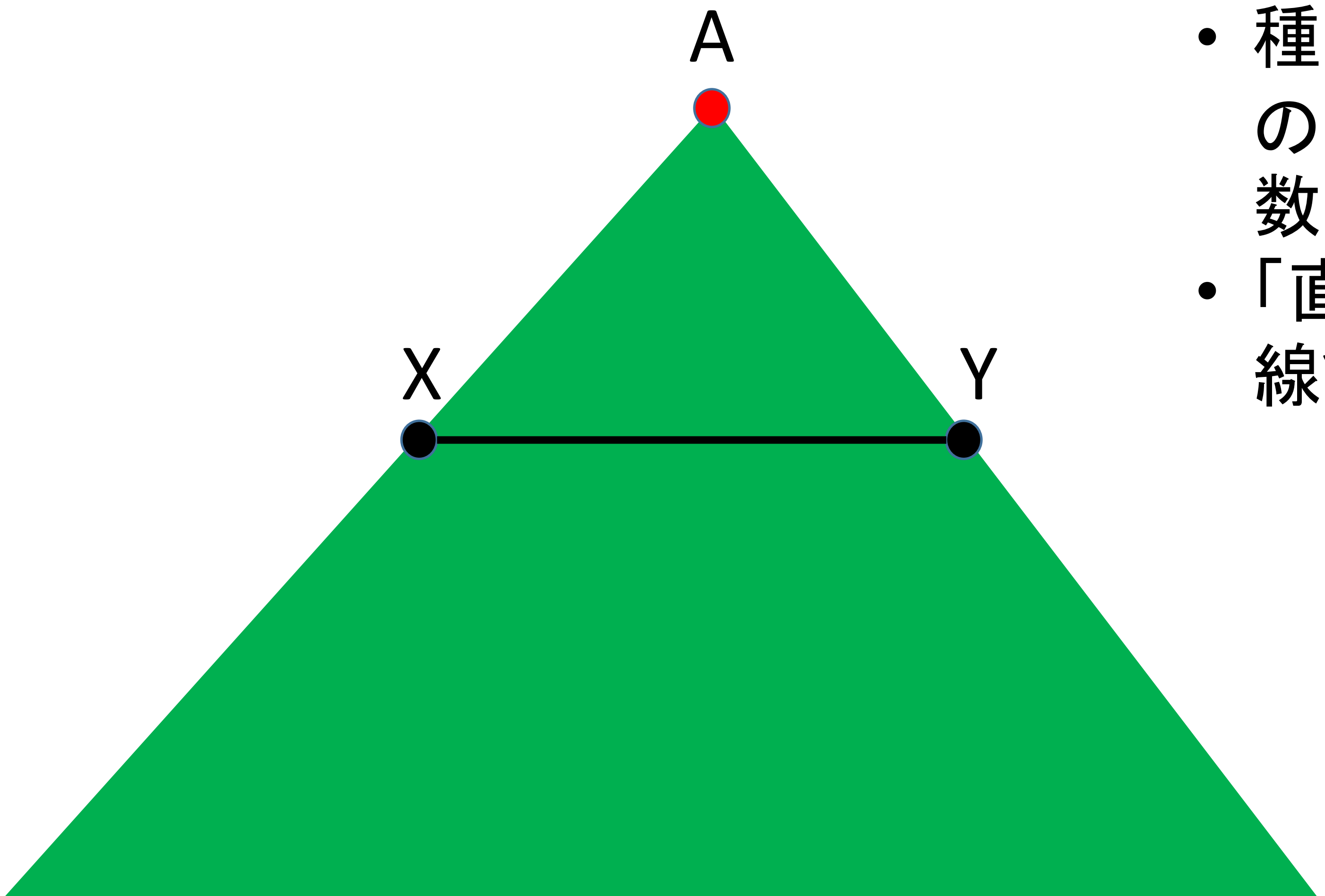
- 注意点
- 座標の値は10^9程度
- 外積の大きさは10^18程度
- →double だと精度が足りない
- →long long か long double を使えばOK

Subtask 2

Subtask 2

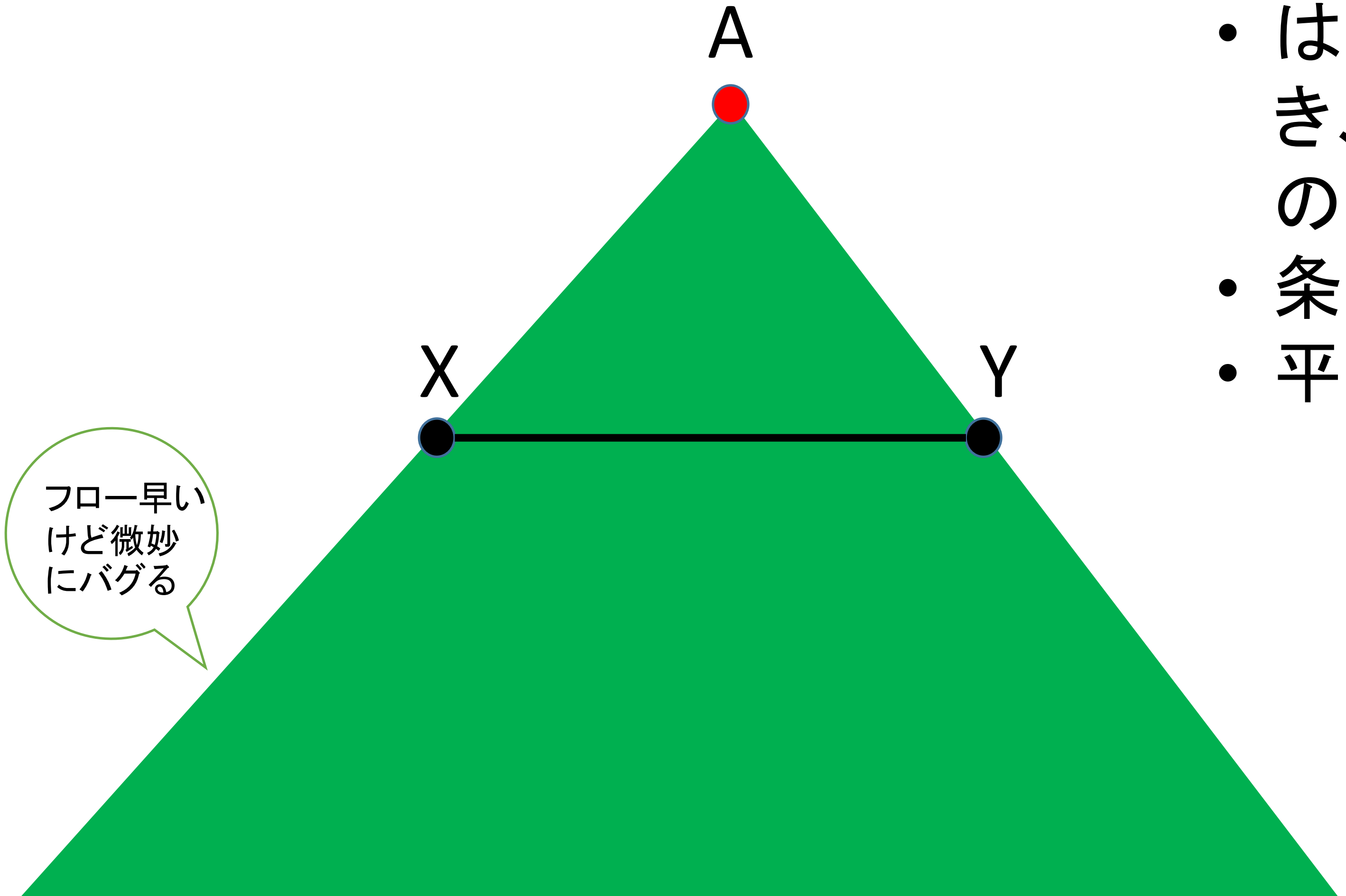
- Q<=100
- 各クエリについてO(N)程度で答えられれば良さそう
- 答えるクエリ（a, b）をfixして考える

Subtask 2



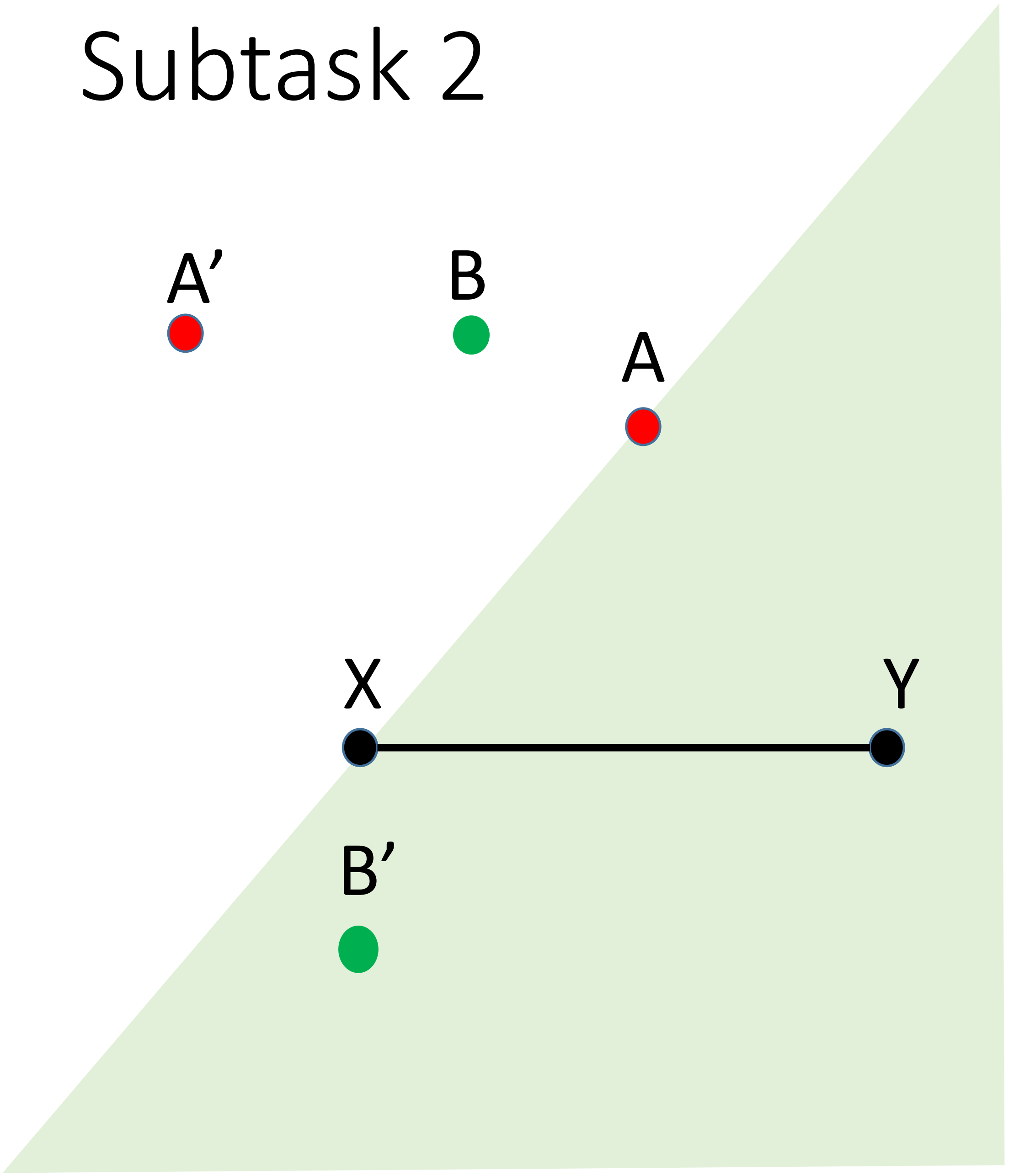
- 種類aの点Aのそれぞれについて、緑の三角形領域内の種類bの点の個数を調べれば良い
- 「直線XAについてYとBが同じ側＆直線YAについてXとBが同じ側」

Subtask 2



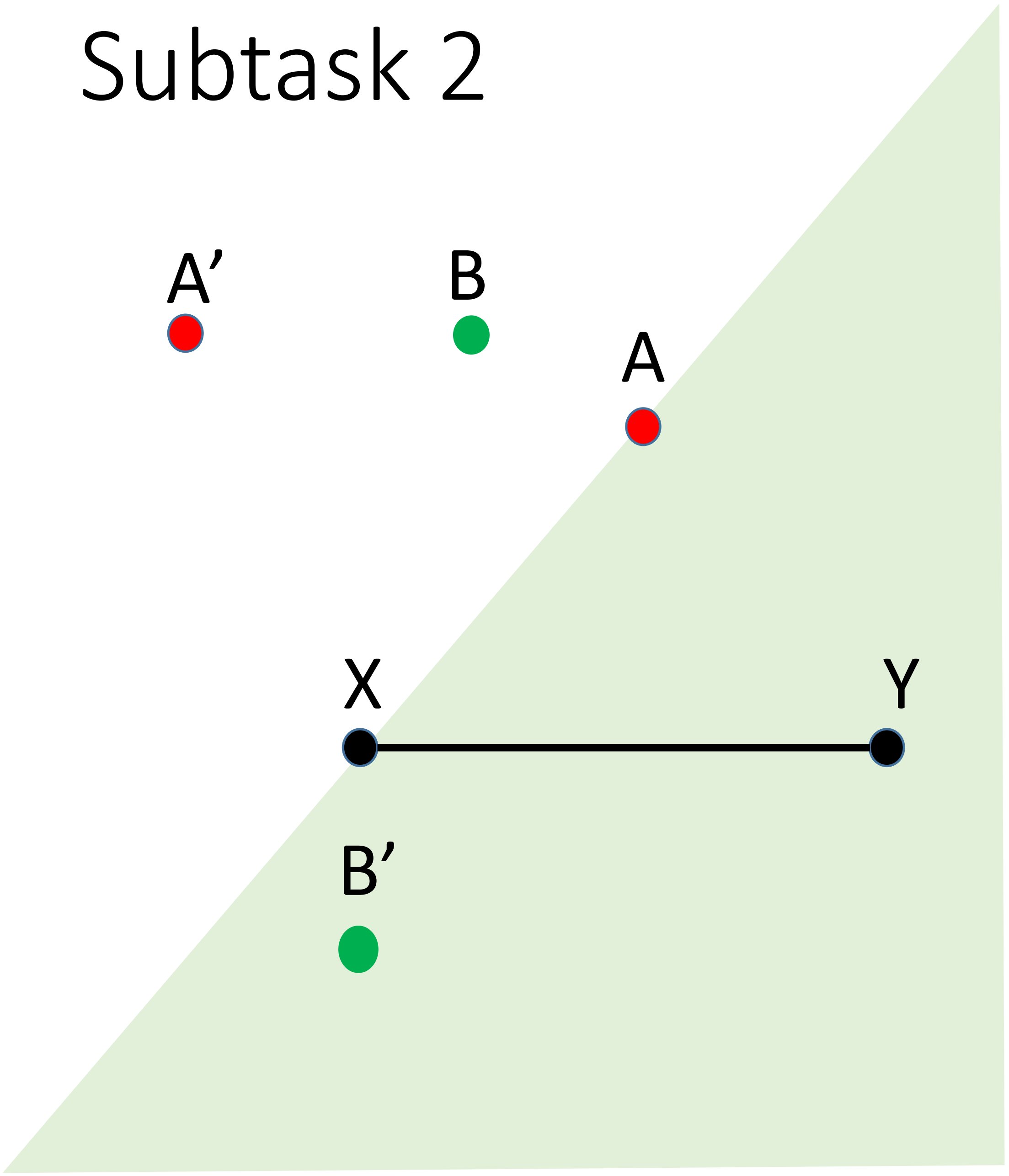
- はじめの条件はXからの直線の傾き、次の条件はYからの直線の傾きのみによる
- 条件はAが動くと連続的に変化
- 平面走査

Subtask 2



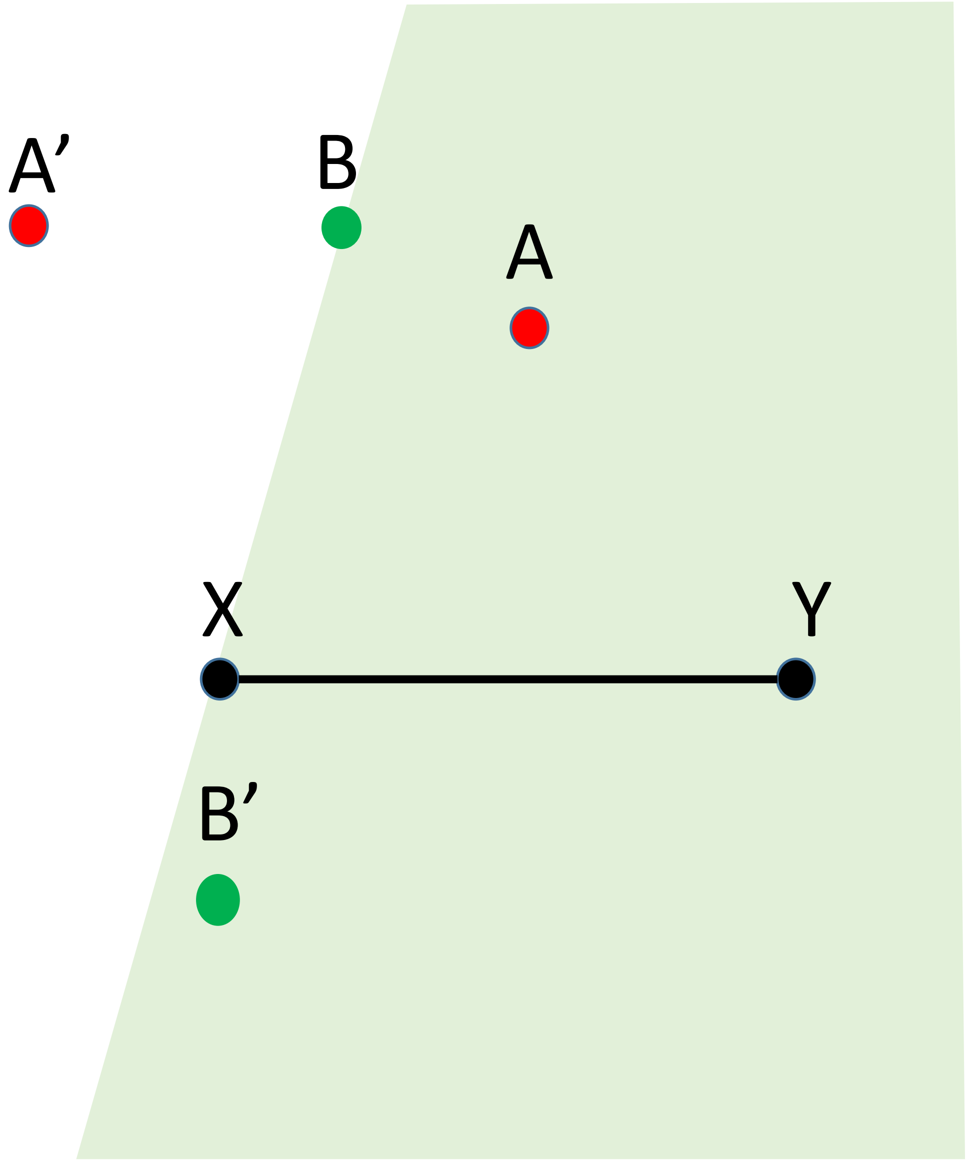
- 点をXからの方向でソート
- 直線XAについてYと同じ側にある点の集合を管理しながらAを動かす

Subtask 2



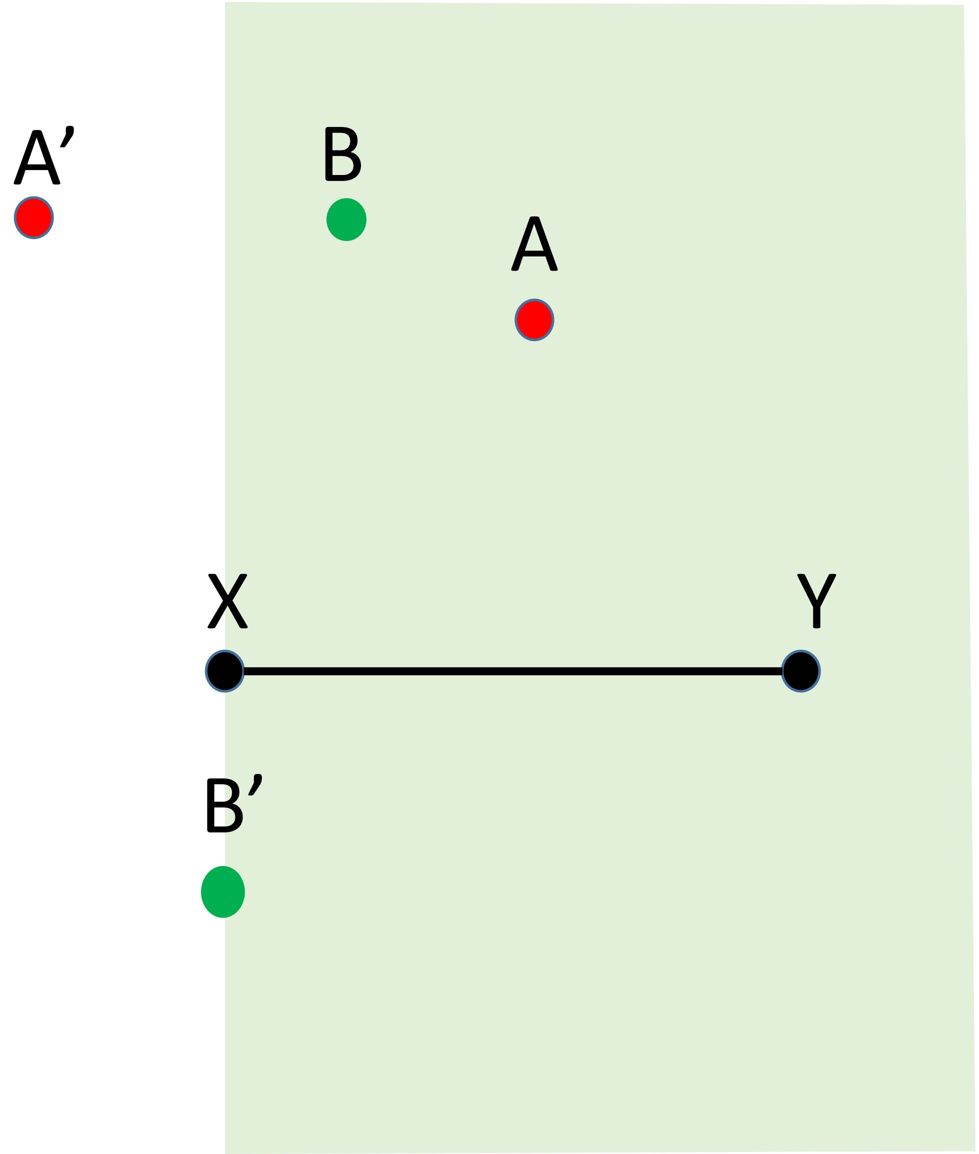
- 点をXからの方向でソート
- 直線XAについてYと同じ側にある点の集合を管理しながらAを動かす
- {B’}

Subtask 2



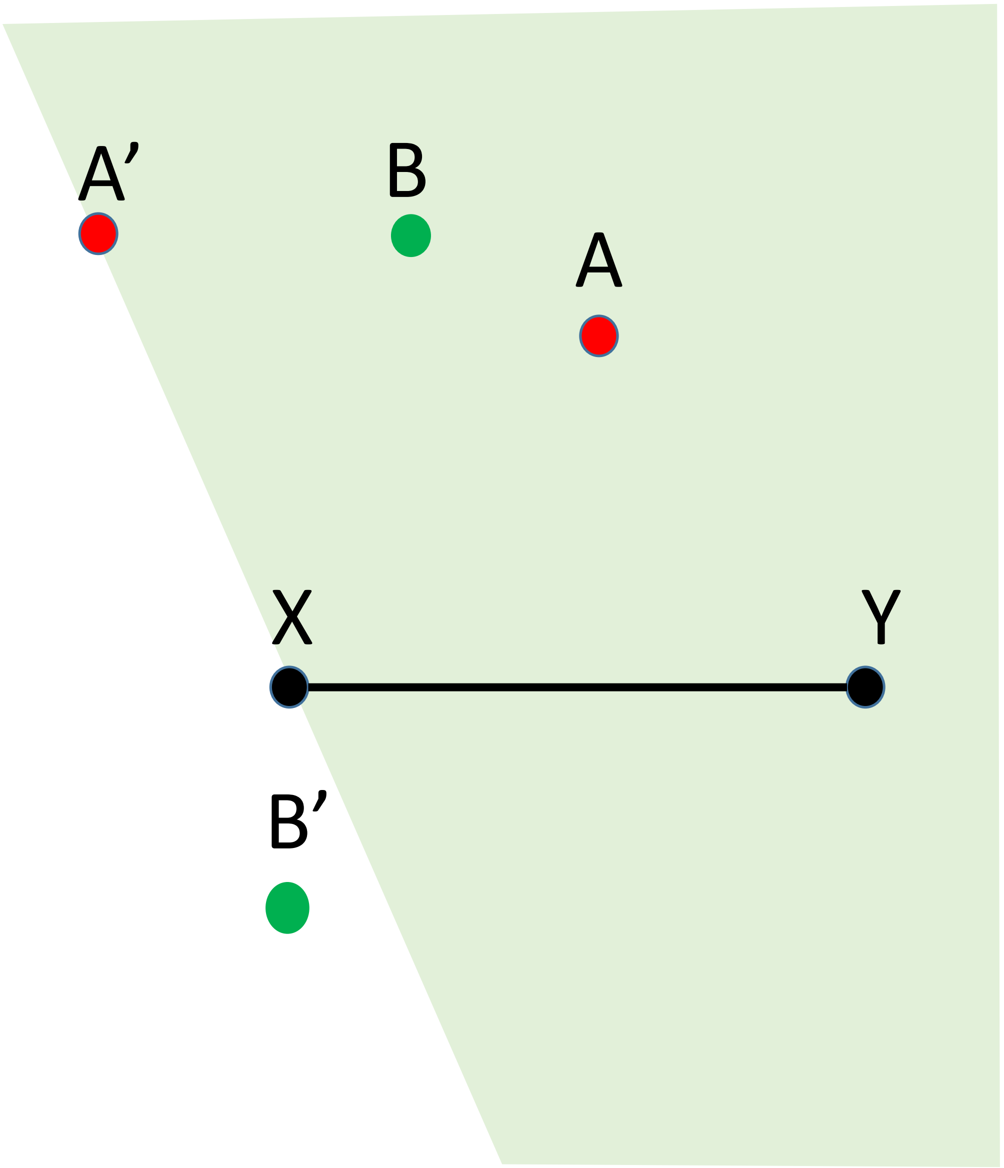
- 点をXからの方向でソート
- 直線XAについてYと同じ側にある点の集合を管理しながらAを動かす
- {B, B’}

Subtask 2



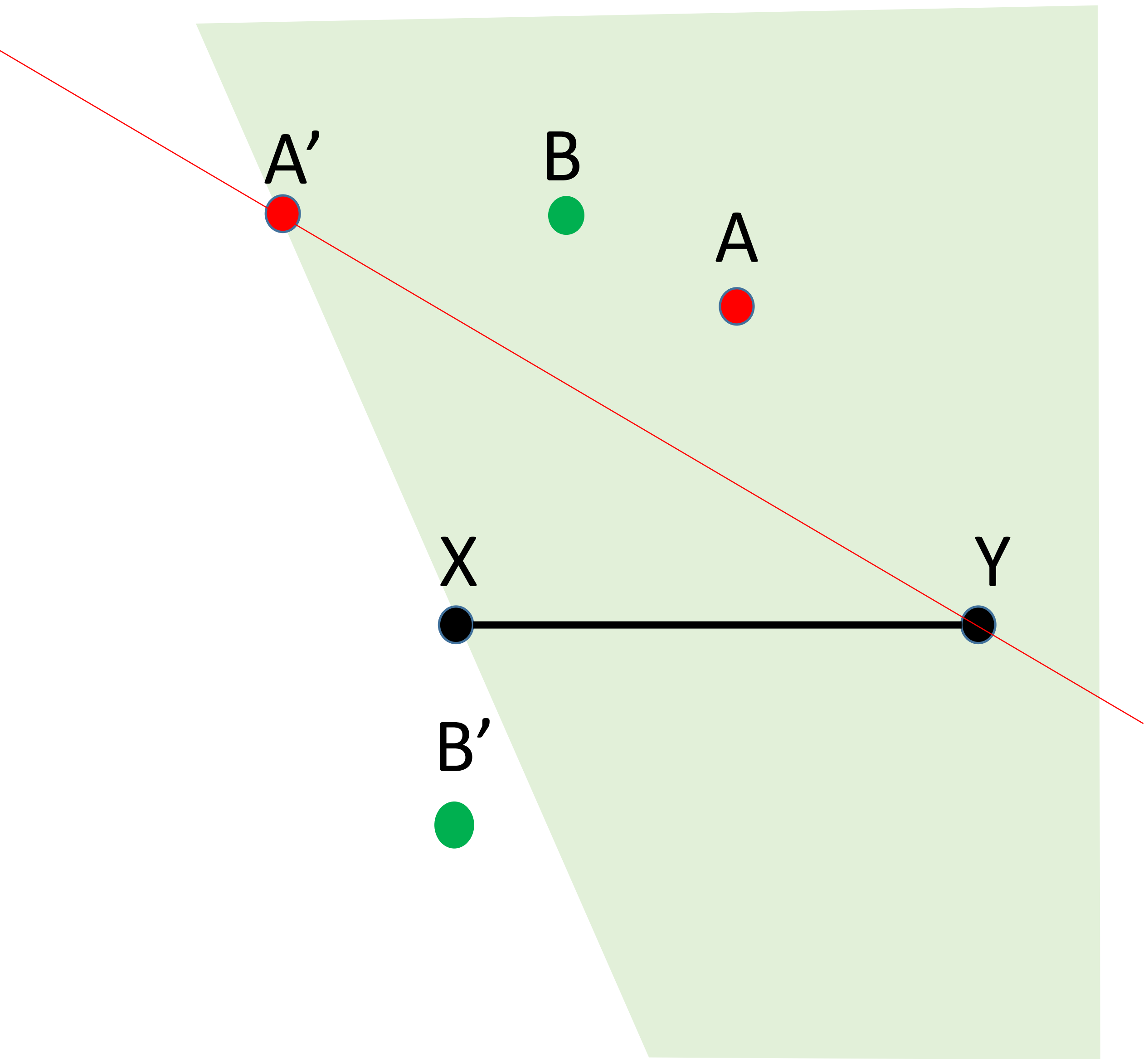
- 点をXからの方向でソート
- 直線XAについてYと同じ側にある点の集合を管理しながらAを動かす
- {B}

Subtask 2



- 点をXからの方向でソート
- 直線XAについてYと同じ側にある点の集合を管理しながらAを動かす
- {B}

Subtask 2



- 点をXからの方向でソート
- 直線XAについてYと同じ側にある点の集合を管理しながらAを動かす
- 種類aの点に到達したら、集合の中でYについての条件を満たすものを数える

Subtask 2

- 次のことができれば良い
- 集合に追加
- 集合から削除
- 集合のうちYからの方向がある範囲のものを数える

Subtask 2

- 次のことができれば良い
- 集合に追加
- 集合から削除
- 集合のうちYからの方向がある範囲のものを数える
- →予め種類bの点をYからの方向でソートしておき、segment treeやBITで管理すれば良い

Subtask 2

- 一つのクエリに対して$O(N \log N)$で解けた

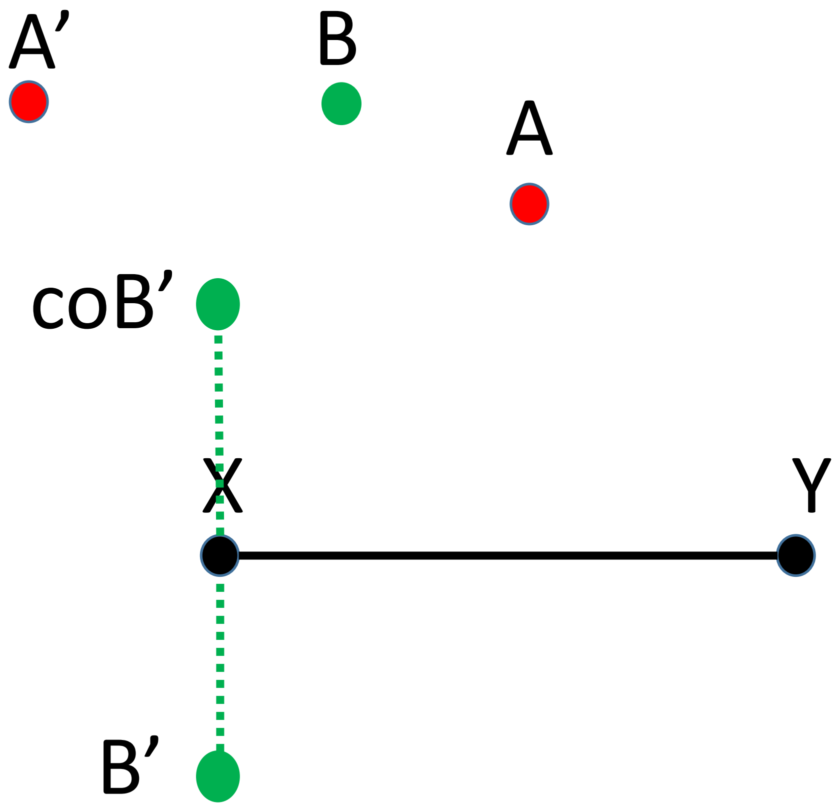
Subtask 2

- 一つのクエリに対してO(N log N)で解けた
- →45点 or 60点
- （うまく実装すればまとめてSubtask 1も解ける）

Subtask 2

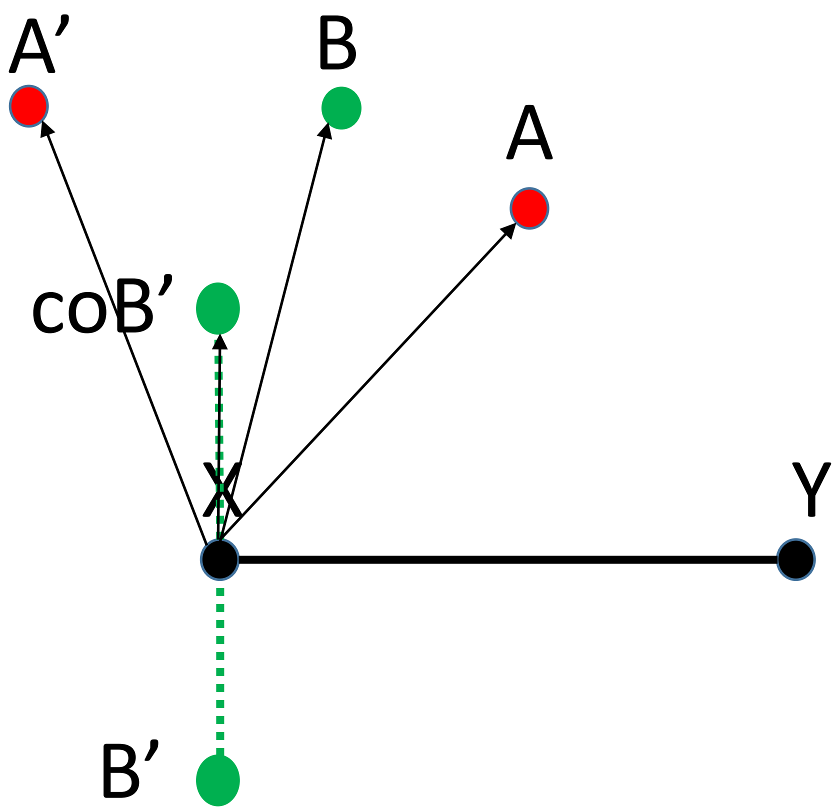
- 実装上の注意
- A, Bそれぞれについて直線XYのどちら側にあるかで場合分けするので面倒
- doubleは精度が足りない
- 誤差が怖いのでatan2で角度を計算しないほうがいい
- はじめに座標を変換して直線XYがX軸に一致するようにすると楽（long doubleを使う場合）。傾きでソートできる

Subtask 2



- 模範的な実装
- はじめにすべての点を直線XYのどちらかの側に集める
- (XについてソートするときはXについての点対称、Yについてソートするときはについての点対称)

Subtask 2

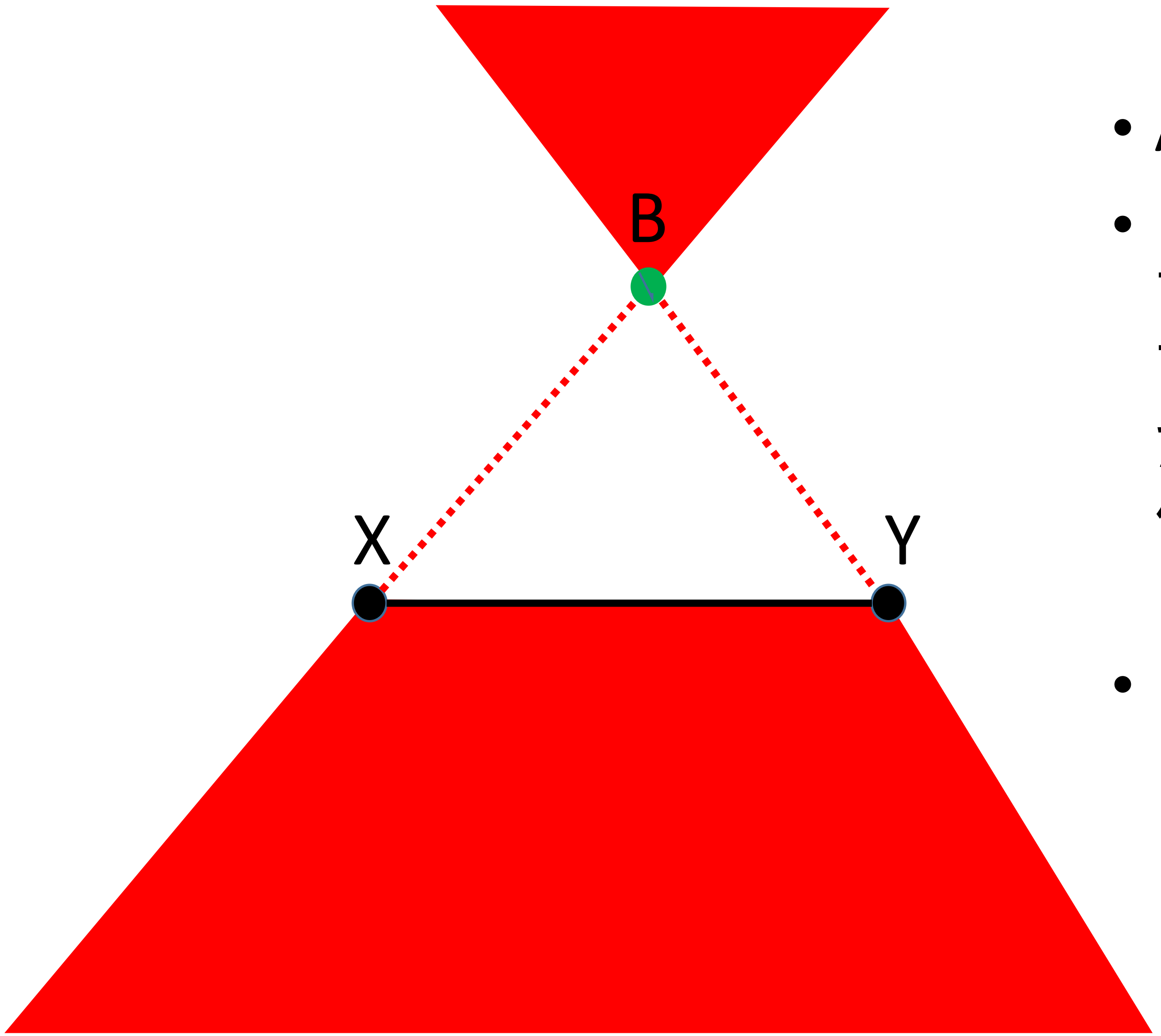


- 模範的な実装
- はじめにすべての点を直線XYのどちらかの側に集める
- (XについてソートするときはXについての点対称、Yについてソートするときはについての点対称)
- 外積により順序が定まる
- P-X<Q-X ⇔ op(P-X, Q-X) > 0

Subtask 2

- BとAの役割を交換しても解ける（伏線）
- Bをfixしたとき、Aがどの範囲にあれば条件を満たすか？

Subtask 2



- Aが赤の領域内にあればよい
- 「直線XYについてBと同じ側＆直線XBについてYと異なる側＆直線YBについてXと異なる側」または「直線XYについてBと異なる側＆直線XBについてYと同じ側＆直線YBについてXと同じ側」
- 同様に平面走査ができる

Subtask 3

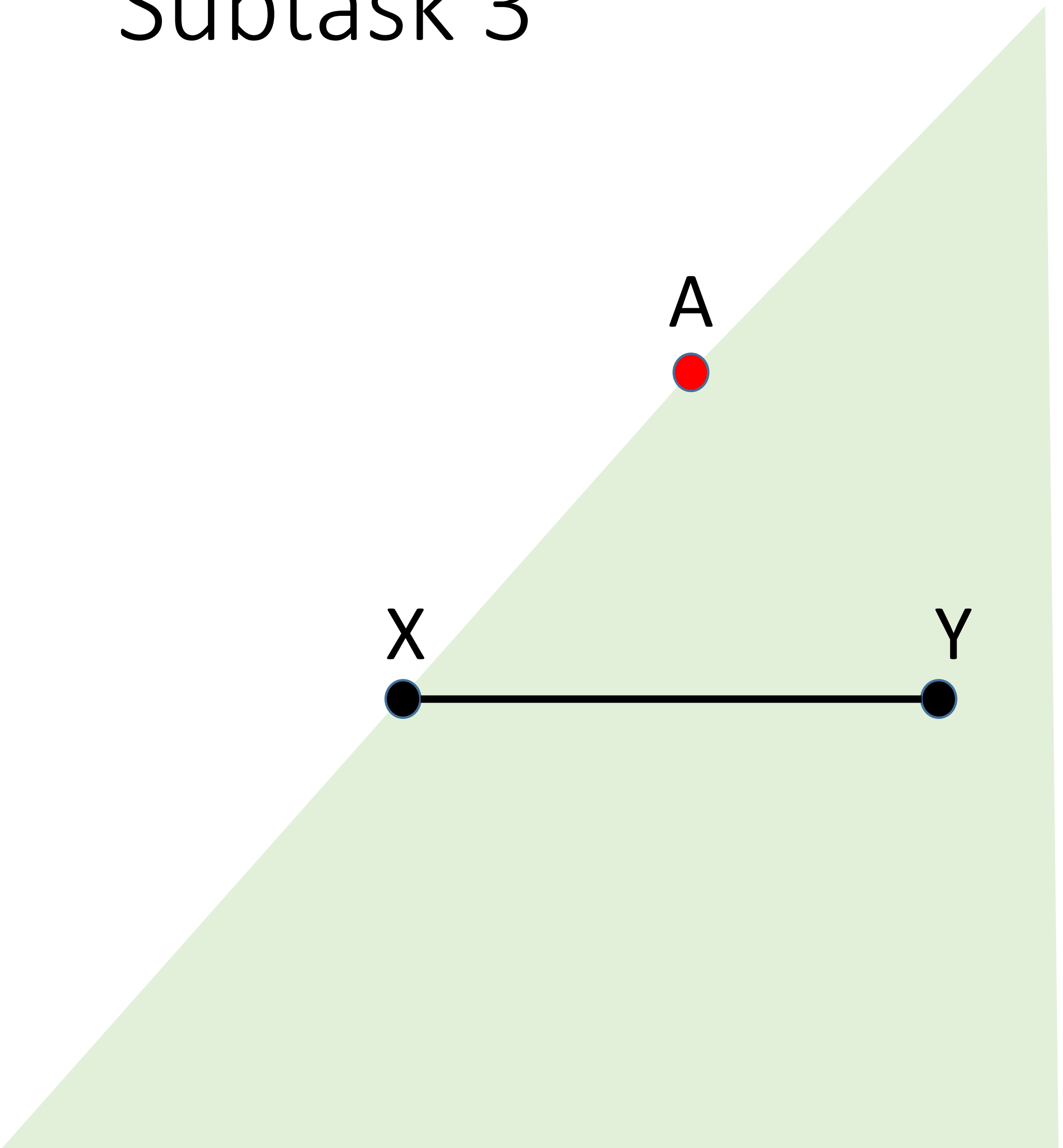
Subtask 3

- Qがでかい
- ひとつひとつのクエリに素早く答えるのは無理そう

Subtask 3

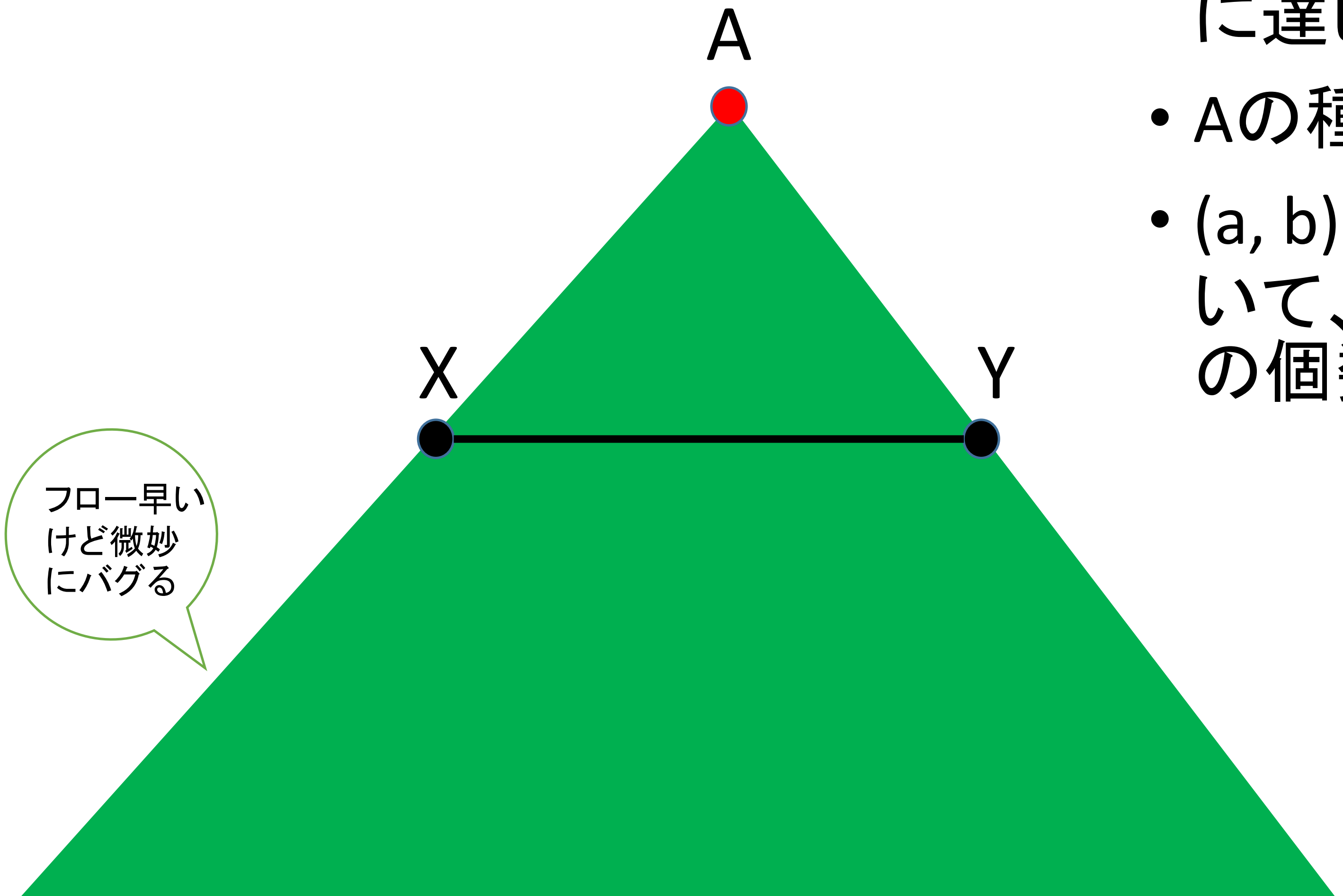
- Qがでかい
- ひとつひとつのクエリに素早く答えるのは無理そう
- クエリを先読みしてまとめて計算するのはできそう
- 予めクエリを全部読んで、適当に分類しておく
- Subtask 2の方法をまとめてやることを考える

Subtask 3



- 前述の平面走査をする
- ただし集合としてすべての種類の点を管理する

Subtask 3



- 任意の種類の点Aについて、そこに達したら次のことを行う
- Aの種類をaとする
- (a, b) の形のクエリそれぞれについて、緑の三角形内の種類bの点の個数を調べる

Subtask 3

- 次のことができれば良い
- 集合に追加
- 集合から削除
- 集合のうちある種類でYからの方向がある範囲のものを数える

Subtask 3

- 次のことができれば良い
- 集合に追加
- 集合から削除
- 集合のうちある種類でYからの方向がある範囲のものを数える
- →予めすべての点を（種類、Yからの方向）でソートしておき、segment treeやBITで管理すれば良い

Subtask 3

- 各点について、
- 「（その点の種類、任意の種類）の形のクエリの数」× O(log N)
- だけ時間がかかる
- 大体のケースで早い
- 点でもクエリでもたくさん登場する種類があるとやばい

Subtask 3

- 各点について、
- 「（その点の種類、任意の種類）の形のクエリの数」× O(log N)
- だけ時間がかかる
- 大体のケースで早い
- 点でもクエリでもたくさん登場する種類があるとやばい
- →そういう種類だけ例外的に処理すればよさそう

Subtask 3

- 種類aで、(a,b)というクエリが√Q個以上存在するものを重い種類、そうでない種類を軽い種類ということにする
- クエリ(a, b)で、種類aが重いものを重いクエリ、そうでないものを軽いクエリということにする
- 軽いクエリについては前述の方法で計算する
- これはO(N√Q log N)でできる

Subtask 3

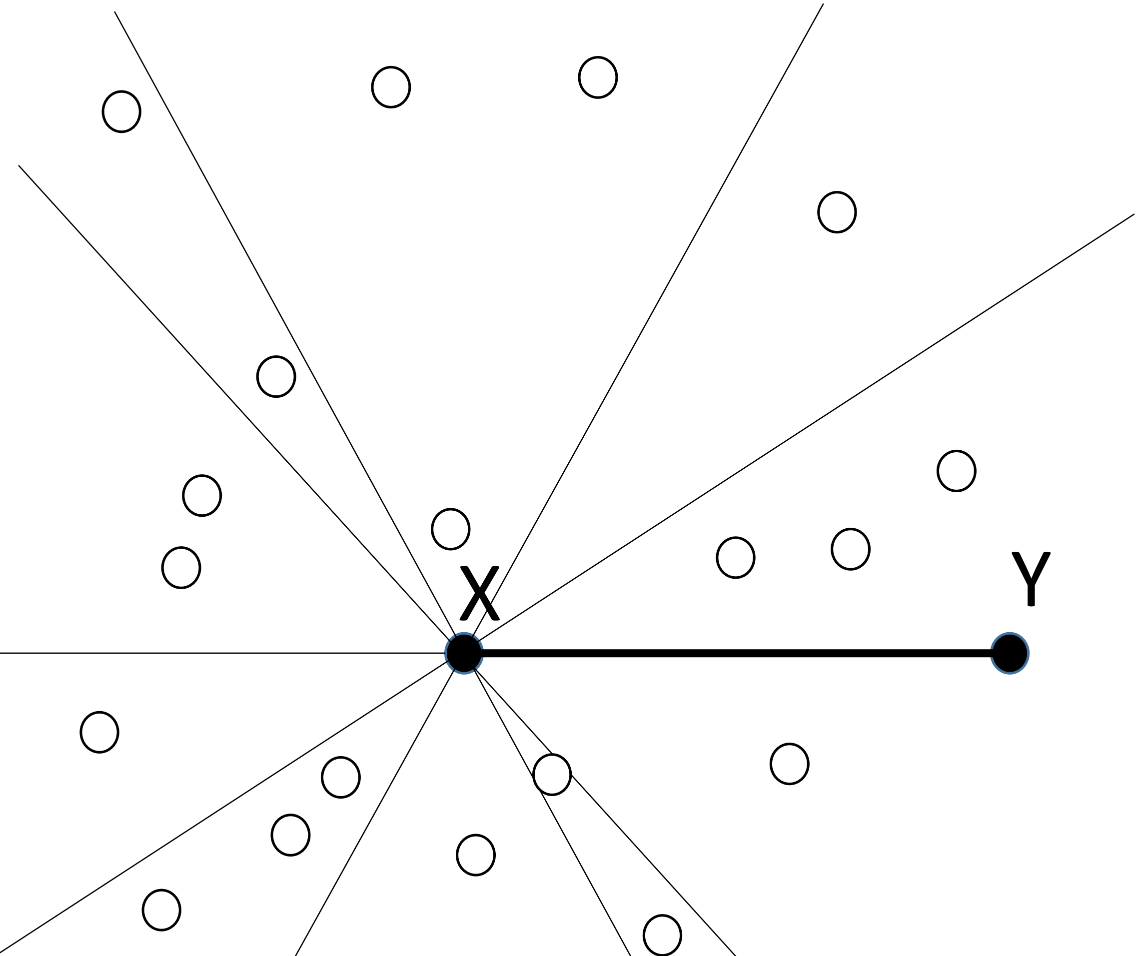
- 重い種類は高々√Q個しかない
- 種類bについて、(a, b)の形の重いクエリは高々√Q個しかない
- 重いクエリについては、BとAの役割を入れ替えて同様に計算する
- これもO(N√Q log N)
- よって全体でO(N√Q log N)

Subtask 3

- 重い種類は高々√Q個しかない
- 種類bについて、(a, b)の形の重いクエリは高々√Q個しかない
- 重いクエリについては、BとAの役割を入れ替えて同様に計算する
- これもO(N√Q log N)
- よって全体でO(N√Q log N)
- →満点！

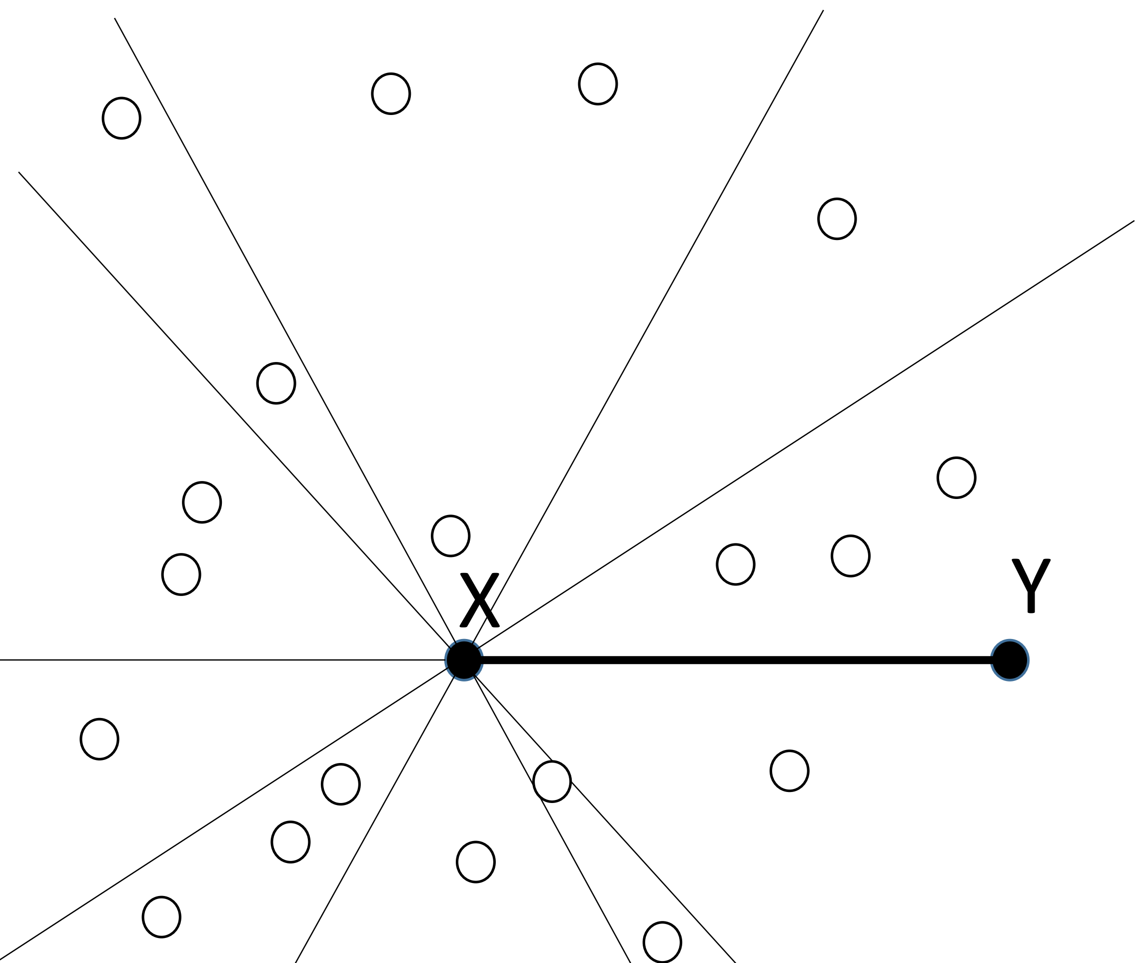
より良い解

より良い解



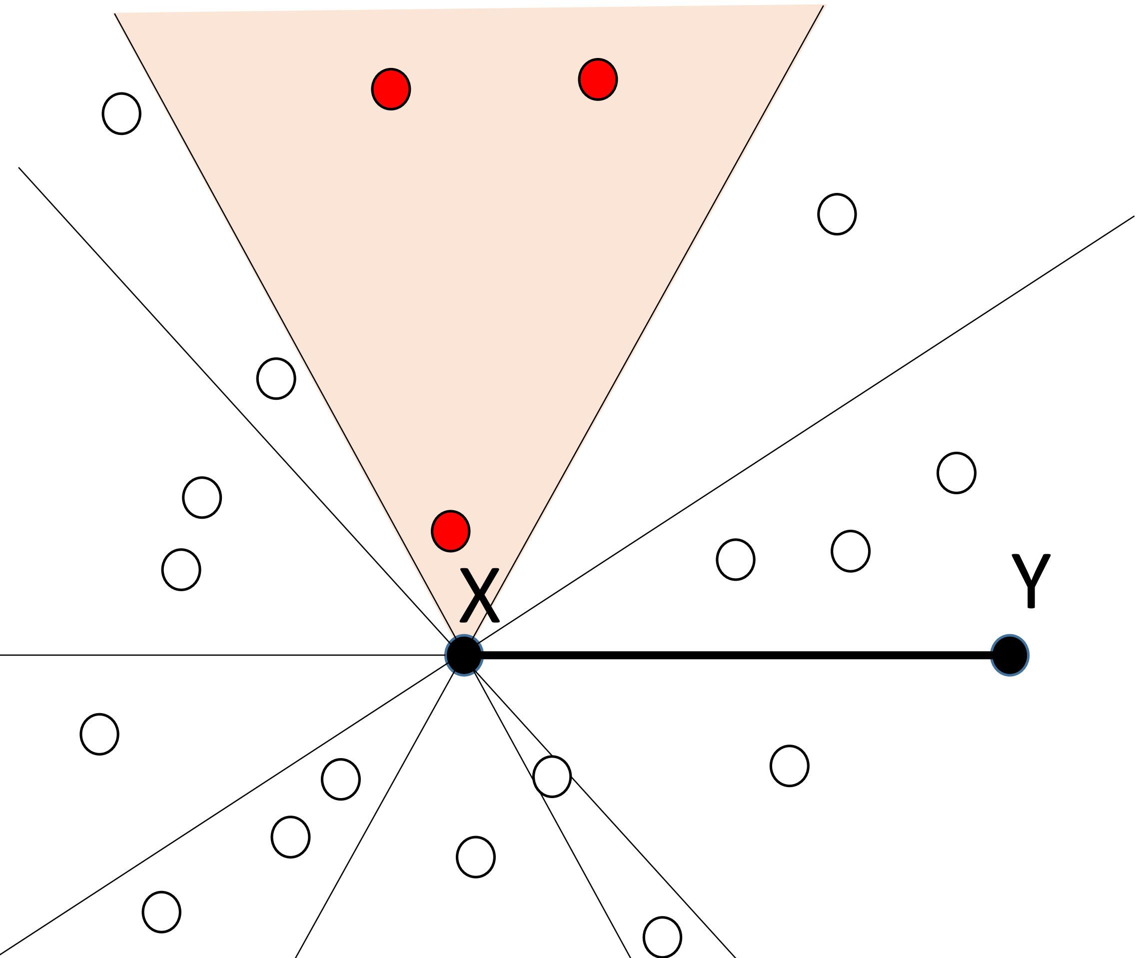
- 実はO(N√N+N√Q)で解ける
- 右図のようにXからの方向でソートして√N個ごとに区切る
- （点対称なふたつの領域は一つの区分とみなすことに注意）
- 図はN=19

より良い解



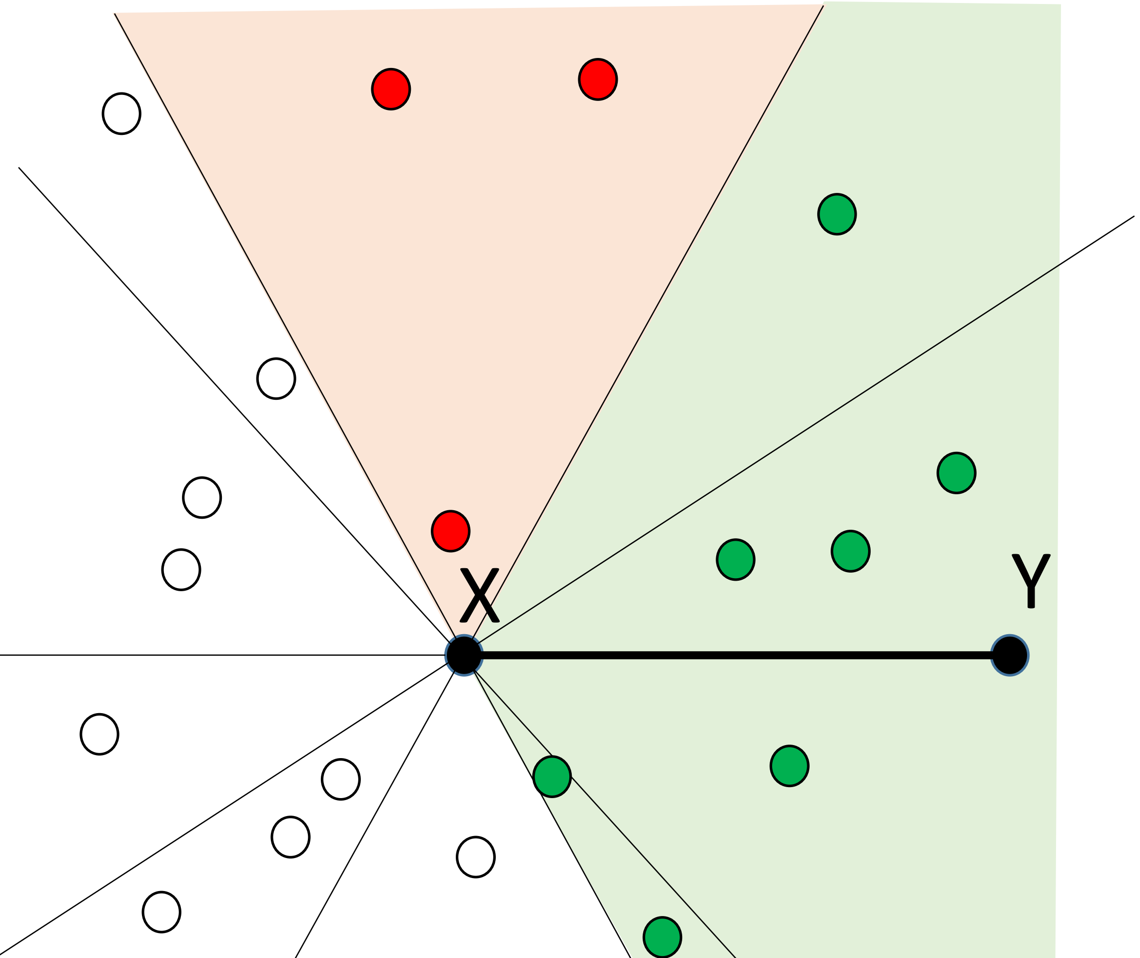
- 二点A, Bが同じ区分に属している場合と違う区分に属している場合に分ける
- 同じ場合
- 考えられる点の組がO(N√N)しかない
- →全部試す

より良い解



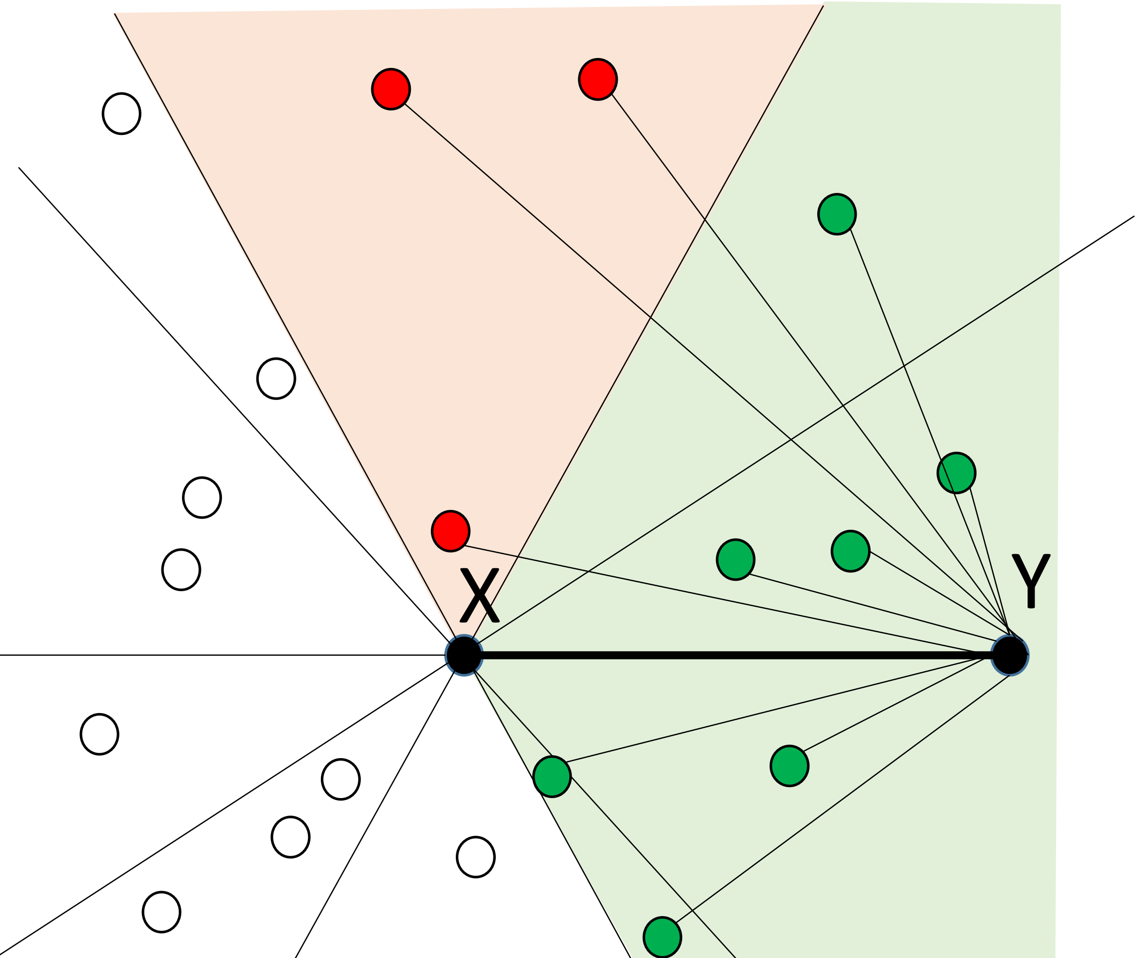
- 違う場合。
- 例えば右図の赤の領域にAがある場合を考える

より良い解



- 違う場合。
- 例えば右図の赤の領域にAがある場合を考える
- Bの候補として考えられるのは緑の領域の点
- これらの点ではXについての条件は自動的に満たされている

より良い解



- これらの点をYからの方向順に見ていけば良い
- 緑の点ではどの種類がいくつ出てきたかをカウントしておく
- 赤の点ではクエリを処理する
- 領域内の点についてまとめて計算できる

より良い解

- 一つのクエリに対しQ(N√N)で計算できる

より良い解

- 一つのクエリに対し$O(N\sqrt{N})$で計算できる
- クエリをまとめて処理すれば$O(N\sqrt{N}+NQ)$

より良い解

- 一つのクエリに対しO(N√N)で計算できる
- クエリをまとめて処理すればO(N√N+NQ)
- →60点

より良い解

- 一つのクエリに対しO(N√N)で計算できる
- クエリをまとめて処理すればO(N√N+NQ)
- →60点
- 前述の平方分割を行えばO(N√N+N√Q)

より良い解

- 一つのクエリに対し$Q(N\sqrt{N})$で計算できる
- クエリをまとめて処理すれば$O(N\sqrt{N}+NQ)$
- →60点
- 前述の平方分割を行えば$O(N\sqrt{N}+N\sqrt{Q})$
- →満点

嘘解法

嘘解法

- Subtask 1の計算量を思い出してみる
- 時間計算量O(N^2)
- N<=30000

嘘解法

- Subtask 1の計算量を思い出してみる
- 時間計算量O(N^2)
- N<=30000
- →定数が軽ければ通る可能性が微粒子レベルで存在している……?

嘘解法

- M×M配列を用意するとMLE
- 点を種類ごとに分ける
- クエリ(a, b)に対し、種類aの点と種類bの点の組に対しすべて試す
- →同じクエリが来ないことから計算量O（N^2+Q）になる

嘘解法

- M×M配列を用意するとMLE
- 点を種類ごとに分ける
- クエリ(a, b)に対し、種類aの点と種類bの点の組に対しすべて試す
- →同じクエリが来ないことから計算量O（N^2+Q）になる
- MLEが回避できた

嘘解法

- 一度の判定をできるだけ早く行いたい
- 前述のように外積を使うと最低でもlong longの掛け算が4回必要
- →厳しい？

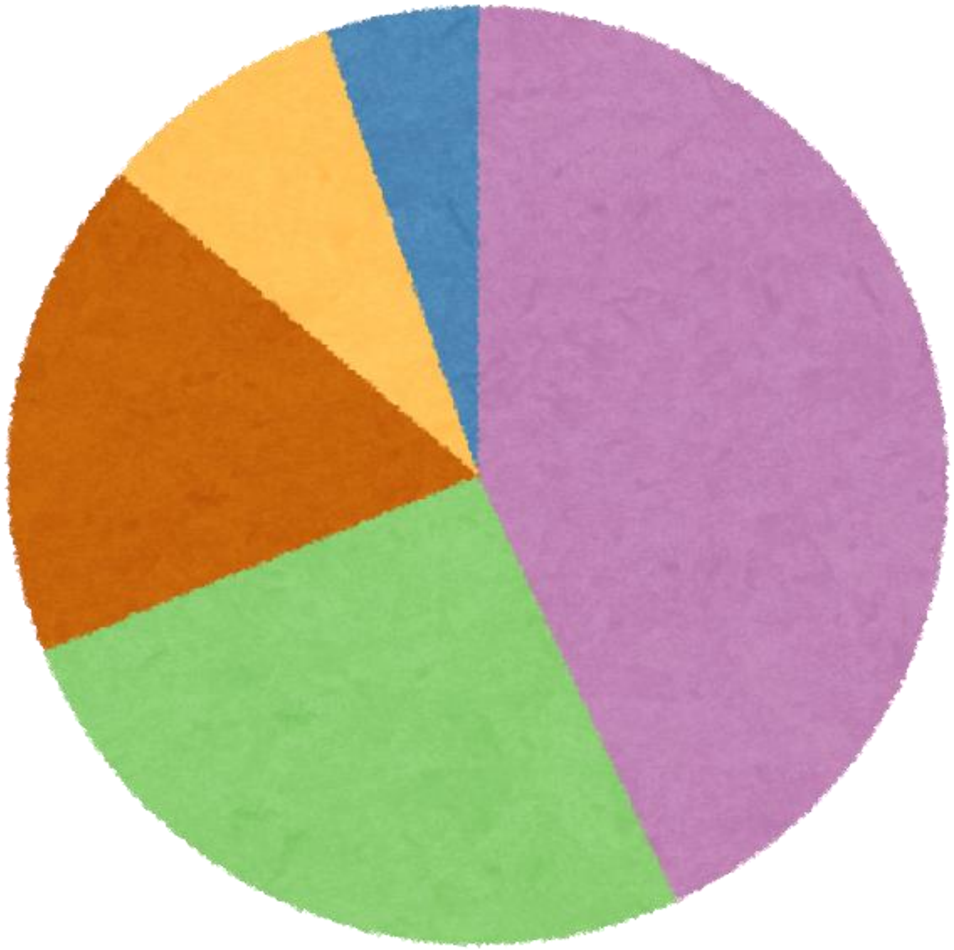
嘘解法

- 一度の判定をできるだけ早く行いたい
- 前述のように外積を使うと最低でもlong longの掛け算が4回必要
- →厳しい？
- 予め全点をXからの方向とYからの方向でソートしておく
- →数の比較だけで判定ができる

嘘解法

- 一度の判定をできるだけ早く行いたい
- 前述のように外積を使うと最低でもlong longの掛け算が4回必要
- →厳しい？
- 予め全点をXからの方向とYからの方向でソートしておく
- →数の比較だけで判定ができる
- →満点？

得点分布



得点分布