

JAG ICPC模擬地区予選2020

F: Mountain View

原案: shora_kujira16

問題文: Darsein

データセット: climpet

解答: climpet, Darsein, hos, tomerun, tsutaj

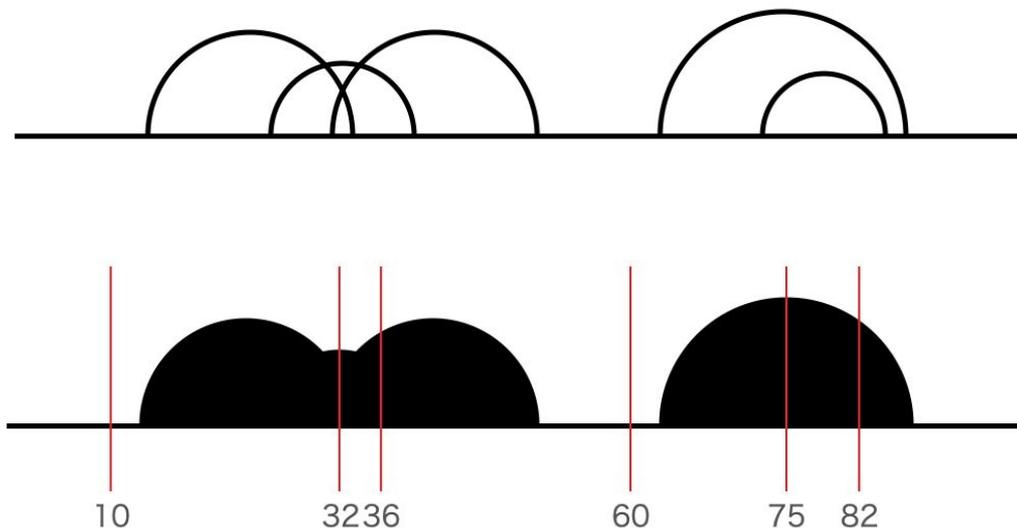
解説: Darsein

問題概要

N個の半円が1列に並んでいる (重なっていることもある)

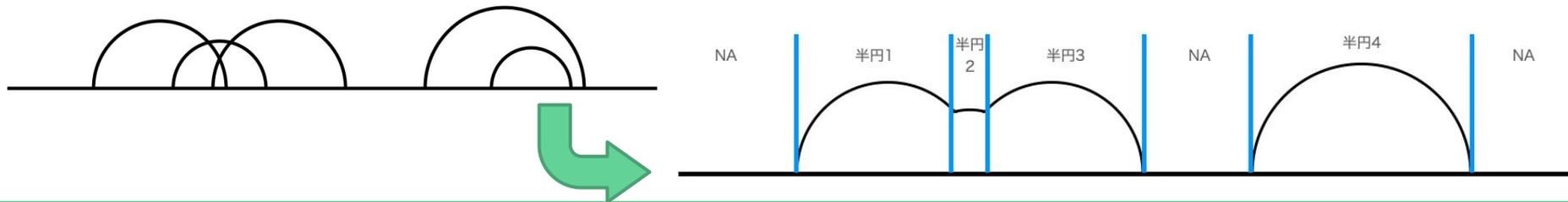
地点がQ個与えられるので、それぞれの地点で最も高い半円の高さを求めよ

制約: $1 \leq N \leq 10^5$, $1 \leq Q \leq 10^5$



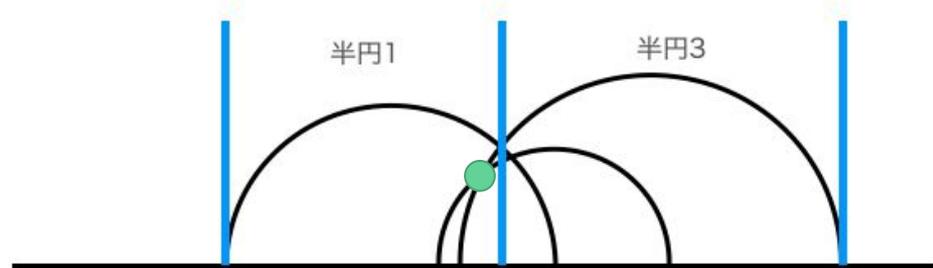
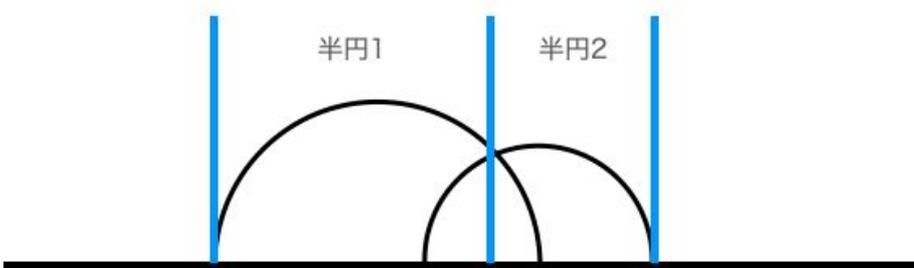
観察

- 直線は高々1つの半円が最大の高さを持つ区間に分割できる
 - 各区間の左端・右端の位置およびその区間で最も高い半円がわかれば、ある位置 x の高さは x を含む区間に対応する半円の x での高さ (なければ0)
- 同じ半円は高々1つの区間にしか含まれない
- 区間 i と区間 j について、 $i < j \Rightarrow$ 区間 i の半円の左端 $<$ 区間 j の半円の左端
 - 左端の昇順に半円を見ていき、いま見ている半円が最も高くなる区間があれば新たに区間を右に追加すればよい
 - 区間は[1つ前の区間の左端, 半円の交点]と[半円の交点, 今見ている半円の右端]に分割される



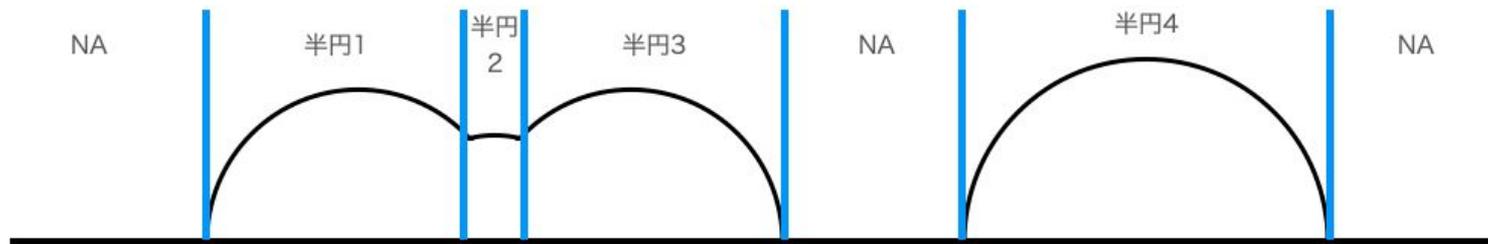
コーナーケース

- より右にある半円により既存の区間がカバーされることがある
 - この場合、1つ前の区間の半円と今見ている半円の交点が、1つ前の区間よりも必ず左にあることを利用して判定できる
 - 既存の区間のうち右側の区間からいくつかしかカバーされないので、右側から順に新しい区間でカバーされるか判定し、カバーされるなら削除する
- 今見ている半円が既存の半円に完全にカバーされるとき、交点を持たないことにも注意



解法: 前処理で区間を作成 + 二分探索

- 前述のように左端の昇順に半円を見ていき、追加/削除を繰り返し区間を作成
- 各クエリに対し、 x がどの区間に含まれるかは二分探索でわかる
- ある半円における位置 x での高さは $\sqrt{r_i^2 - (c_i - x)^2}$ で計算可能
 - 範囲外の場合、高さは0



計算量

- 区間の作成 (前処理):
 - 半円を左端の位置でソート: $O(N \log N)$
 - 区間の作成: $O(N)$
 - ある1つの半円は追加も削除も高々 1回までしかされないのので、追加 / 削除の回数は $O(N)$
 - 削除/追加の判定 (= 交点の計算): $O(1)$
 - 削除/追加の操作: スタックなどのデータ構造を用いて $O(1)$
- クエリの処理:
 - Q 個のクエリ各々に対して、
 - x を含む区間を二分探索して $O(\log N)$
 - 高さの計算に $O(1)$
 - クエリ全体で $O(Q \log N)$
- 全体で $O((N+Q) \log N)$
 - クエリをソートすると全体で $O(N+Q)$ で答えることもできる。この場合 $O(N \log N + Q \log Q)$

統計

Acceptances / Submissions

- 23 / 79 (29.11%)

First Acceptance

- __KING__ (00:26)