

# 必須問題

二次元上の点集合  $(x_i, y_i)(i = 0 \dots n - 1)$  が与えられたとする ( $n \geq 2$ )

二点間  $p = (x_i, y_i), q = (x_j, y_j)$  の距離  $D(p, q)$  を以下のようなユークリッド距離と定める。

$$D(p, q) := ((x_i - x_j)^2 + (y_i - y_j)^2)^{1/2}$$

(1) 点集合が与えられた時、距離が最小となるペアの距離を返す関数を実装しなさい

(2) 矩形  $[x_{min}, x_{max}] \times [y_{min}, y_{max}]$  中に含まれる点数を報告する関数を実装しなさい。なお、矩形の境界線上の点も報告対象とする。

(3)  $D(p, q) < r$  であるとき、点  $p$  から点  $q$  は  $r$  で到達可能と呼ぶ。また、点  $p$  と  $q$  が正の実数  $r > 0$  で遷移可能とは、ある正整数  $k$  と点集合中に  $k$  個の点列,  $n_1 = p, n_2, \dots, n_k = q$  が存在し、 $n_i$  と  $n_{i+1} (i = 1 \dots k - 1)$  が到達可能であるとする。点集合中の二点が与えられた時、二点が遷移可能かどうかを判定し、遷移可能なら true, 不可能なら false を返すプログラムを書いてください。

# オプション問題

(上記問題が解けた場合のみ解いてください。必ずしも解く必要はありません。)

(4) 点集合の数を1億として、各点が  $[-1000, 1000] \times [-1000, 1000]$  中にランダムに分布している場合に、 $r = 100, 10, 1, 0.1, 0.01, 0.001$  とした場合の(3)の処理時間を測定してください。

単に解けたかどうかだけでなく、例えば以下のような点から判断します。

- 計算量は少ない方がよい
- メモリ使用量は少ない方がよい
- 近似解であれば、その近似がよいほうがよい
- ユニークなアイデアに基づいているのがよい
- ソースコードの品質は良いのがよい。ただし品質とは何かについては各自考える
- プログラムが正しいかどうかをチェックできている方がよい

以下のプログラミング言語のいずれかを利用してください。

- C, C++, Java, Ruby, Python, JavaScript