

ICPC模擬国内予選2023

F: 井の中の蛙

原案: climpet

問題文: darsein

データセット: climpet

解答: beet, climpet, darsein, ei1333, hos, kotatsugame, mtsd, tatyam

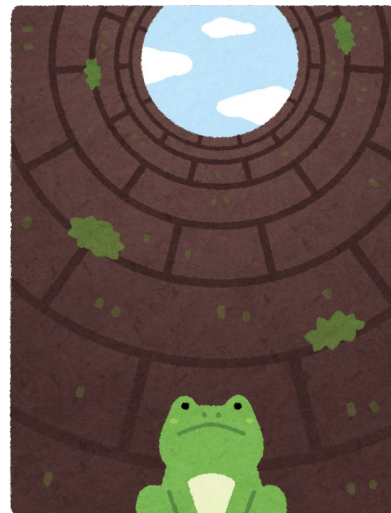
解説: climpet

問題概要

- $h \times w$ の二次元配列 $(s_{r,c})$ が与えられる。
- 各 $d = 1, \dots, h + w - 2$ について、次の条件を満たす要素の個数を求めよ。
 - マンハッタン距離 d 以内に、その要素自身より大きい要素が存在しない。

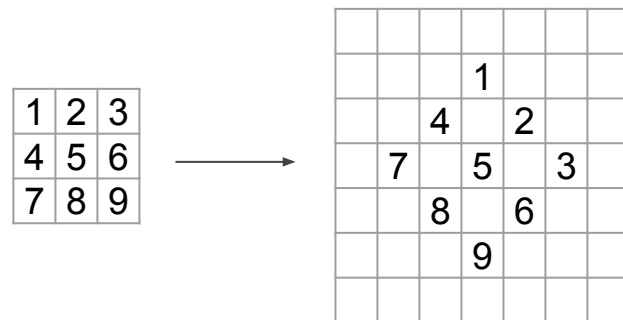
制約

- $h, w \leq 1000$
- $s_{r,c} \leq 10^9$



マンハッタン距離の常套手段

- 入力の配列を 45° 回転させたものを考える。



(空欄の部分は 0 などで埋める)

マンハッタン距離の常套手段

- 入力の配列を 45° 回転させたものを考える。
- このとき、次の 2 つが等しくなる。
 - 回転前の座標系における、ある 2 要素間の マンハッタン距離
 - 回転後の座標系における、ある 2 要素間の チェビシェフ距離
- 例えば、図のオレンジの四角形は、要素 5 から距離 1 以内の領域を表す。
- ある位置からチェビシェフ距離一定の領域は、辺が軸に平行な正方形となる。

解法によっては元のマンハッタン距離より扱いやすくなる場合がある。

| | | |
|---|---|---|
| 1 | 2 | 3 |
| 4 | 5 | 6 |
| 7 | 8 | 9 |



| | | | | | |
|---|--|---|---|---|---|
| | | | | | |
| | | | 1 | | |
| | | 4 | 5 | 2 | |
| 7 | | 8 | 5 | 6 | 3 |
| | | | 9 | | |
| | | | | | |

writer 解

- 45° 回転後の各位置 (r, c) について、次の条件を満たす距離の最大値 d を、二分探索で求めることを考える。
 - (r, c) からチェビシェフ距離 d 以内に、 $s_{r,c}$ より大きい要素が存在しない。
- このためには、次の命題の真偽を判定できればよい。
 - (r, c) を中心とする、一辺の長さ $2d + 1$ の正方形領域について、その領域内の要素の最大値は $s_{r,c}$ に等しいか？
- つまり、二次元配列に対する RMQ に帰着できる。ただし、クエリは常に正方形領域とする。

writer 解

- さらに、以下の事実を用いる。
 - 任意の正方形領域は、一辺の長さ 2^k の正方形4つでちょうど覆うことができる。
(そのような非負整数 k が存在する)
- そこで、すべての位置および非負整数 $k (\leq \log_2 (h + w))$ について、次の値を予め求めておく。
 - その位置を左上とする一辺の長さ 2^k の正方形領域内の、要素の最大値
 - いわゆる Sparse table を二次元に拡張したものと言える。
- RMQ では、このうち 4 箇所を参照し、それらの最大値を求める形となる。
 - 場合によっては、元の入力から「はみ出た」部分を参照することもあることに注意。
- 計算量は全体で $O((h + w)^2 \log (h + w))$ 時間でできる。

その他の解法

- 他にも、様々な計算量の解法が存在する。
 - 45° 回転後、大きい要素から順番に処理する。処理済みの要素の位置を二次元のBITなどで管理することで、ある二次元領域内の処理済み要素数がゼロであるかを判定できるようにする。二分探索も含めて全体で $O((h+w)^2 \log^3(h+w))$ 時間。
 - 大きい要素から順番に、BFS を計 hw 回行う。このとき、各位置における距離の値は $h+w$ 未満の非負整数である。したがって、単純な枝刈りにより、各位置での距離の値の更新回数は $h+w$ 回未満に抑えられる。全体で $O(hw(h+w))$ 時間。
 - なんでもいいから正しい解法を書いて、実行終了までひたすら待つ。
 - 今回のデータセットの場合、愚直解でも 30分くらいあれば実行が終わるようなので、とりあえず書いてみるのも良いでしょう。

ジャッジ解

- climpet (C++, RMQ): 73行, 1535 bytes
- darsein (C++, BIT): 154行, 3511 bytes
- hos (C++, BIT): 84行, 2412 bytes
- kotatsugame (C++, BIT): 91行, 1554 bytes
- ei1333 (C++, BFS): 63行, 1507 bytes
- tatyam (C++, BFS): 47行, 1302 bytes
- beet (C++, TLE): 75行, 1492 bytes
- mtsd (C++, TLE): 144行, 4643 bytes

統計情報

- AC / trying teams
 - 17 / 18
- First acceptance
 - Speed Star (86:25)