

ABC170-F 解説

ホスフィン @mine691

July 2, 2020

問題文

$W \times H$ のグリッドが与えられる。(移動不可能なマスもある) (x_1, y_1) から出発して, (x_2, y_2) に移動することを考える. このとき, コスト 1 で上下左右 1 方向に最大 K マスまで移動できる. 移動コストの最小値を求めよ.

制約

- 入力はすべて整数
- $1 \leq H, W, K \leq 10^6$
- $1 \leq H \times W \leq 10^6$

簡単な考察

愚直にやってみると, $O(HWK)$ かかりそうでダメ
「進行方向を変えたとき」と「 K 以上進んだとき」でコストが増加する
 $\text{dist}[x][y][\text{direction}]$ の形で (最短経路, 進んだ回数) を求めると良さそう (ダイクストラしたい気持ちが抑えきれない)

解法例

- 拡張ダイクストラ $O(HW \log(HW))$
- 01-BFS $O(HW)$
- 枝刈り (?) BFS $O(HW)$
- noshi 式区間に辺を貼るテク $O(HW \log K)$
- set で頑張るやつ

今回は上 3 つを説明する.

拡張ダイクストラとは？

正確には、「拡張（されたグラフ上で）ダイクストラ」のこと
ダイクストラをするとき、最短経路と一緒に別の情報を持っ
ておくことで、色々な量を一緒に求めることができる
priority_queue を使うためには、「優先度」がついたものでな
いといけない
本問題では、

（最短経路長、ある方向に何回連続で進んだか）

を優先度にする。

拡張ダイクストラ解法

説明がめんどくさいから コード を見てくれ

01-BFS 解法

グリッド上で、辺のコストが 0 or 1 の場合, 01-BFS というアルゴリズムが使える

deque (両端キュー) を使うと $O(HW)$

01-BFS については「01-BFS のちょっと丁寧な解説」で検索！
コード

枝刈り BFS 解法

実は, 愚直でも工夫すれば通る
 K 進むとき,

- 自分より大きいコストがあれば無視して進む
- 自分以下のコストが現れたら終わり

をすれば, 各マスは高々2回 or 4回しか見ない
全体として $O(HW)$ でできる
コード

その他の解法

noshi 式区間に辺を貼るテク

set で頑張るやつ：未到達のマスを set で持つ. 実装で破滅しがち.