

JOI 2024/2025 春季トレーニング Day 1

ビ太郎の旅 2

解説担当: 兼下航輔(potato167)

問題概要

- $H \times W$ のマス目のマス (i, j) には整数 $T_{i,j}$ が書かれている。
- マス (a, b) からマス (c, d) に1回の**ハイジャンプ**で移動できることは以下と同値
 - マス (a, b) からマス (c, d) に、上下左右に隣り合う $L + T_{a,b}$ 以下の数字が書かれたマスのみを辿って移動できる。
- Q 個のクエリを処理する。
 - マス (A_k, B_k) からマス (C_k, D_k) に移動するのに何回の**ハイジャンプ**が必要か

小課題 1 : $H \times W \leq 300$

- 全てのマスの組 $((a, b), (c, d))$ について、マス (a, b) からマス (c, d) に 1 回のハイジャンプで移動できるかを求める。
 - 始点を固定すると BFS など全ての終点に対して 1 回のハイジャンプで移動か否かが $O(HW)$ で求まる
- その後、ワーシャルフロイド法をすると、全てのマスの組に対する答えが求まる。
 - 時間計算量は $O((HW)^3)$
- クエリは $O(1)$ で処理

小課題 2: $H \times W \leq 3000$

- マス (A_k, B_k) から 1 回目のハイジャンプで行くべきマスは以下のいずれかに限定して良い。
 - (C_k, D_k)
 - マス (A_k, B_k) から 1 回のハイジャンプで行けるマス (i, j) の中で、 $T_{i,j}$ が最も大きいマス。← (A_k, B_k) の**ベストな行き先**と呼びます。
- 最後の 1 回以外は行けるマスの中で $T_{i,j}$ が最も大きいマスに移動し続けるのが最適。

クエリ前計算

- 全てのマス (a, b) に対して、BFS 等で以下の 3 つを求めておく
 - (a, b) から 1 回のハイジャンプで移動可能なマス
 - 移動可能なマスの中で $T_{i,j}$ が最も大きいマス (i, j)
 - マス (a, b) から始めて、 $T_{i,j}$ が最も大きいマス (i, j) へハイジャンプを x 回 ($1 \leq x \leq HW$) 繰り返したときに辿りつくマス
- これらは $O((HW)^2)$ で求められる。

クエリ処理

- (A_k, B_k) から (C_k, D_k) まで x 回で辿り着けるかを判定問題を解く。
 - 前計算の結果を用いて $O(1)$ で求められる。
- x を二分探索すると、クエリごとに $O(\log(HW))$ で求まる。

小課題 3: $H \times W \leq 150000$ 始点固定

- 今まで通り全ての始点に対して行ける場所を列挙しては間に合わない。
- BFS していたところを priority_queue にすることで、全ての答えを $O(HW \log(HW))$ で求められる。

例

- マス (1, 1) から 1 回のハイジャンプでいけるマスと、それに隣接するマスを管理する。
- 次は $57 + L$ 以下のマスを隣接するマスから探す。
- 辿り着けるマスに隣接するマスを `priority_queue` で管理すれば良い。

53	55	51	49
56	60	89	45
54	57	92	43
96	99	95	92

小課題 4: $H \times W \leq 150000, Q \leq 150000$

以下の二つを高速に処理したい。

- 前計算で、マス (a, b) のベストな行き先を全ての (a, b) に対して求める。
- クエリを回数に対する二分探索で処理する。

前計算

- マス (a, b) から 1 回のハイジャンプで行けるマス (i, j) の中で、 $T_{i,j}$ が最も大きいマスを全ての (a, b) に対して求める。
- 全てのマスについて個別にやる余裕はないので、同時に求める。
- Union Find を用いると時間計算量が改善できる。

前計算

Q. マス (a, b) から 1 回のハイジャンプで行けるマス (i, j) は？

A. 書かれている数が $T_{a,b} + L$ 以下の隣り合うマスの組を全て辺で結んだときに、 (a, b) と連結であるもの

例

- T が左のマス目で表され、 $L = 3$ のときを考える。

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

例

- $(3, 3)$ の周りを見る。
- 変化なし。

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

例

- $(3, 4)$ の周りを見る。
- $(3, 3)$ とマージされる。

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

例

- $(1, 1)$ の周りを見る。
- 変化なし。

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

例

- $(2, 4)$ の周りを見る。
- $(3, 4)$ とマージされる。
- $T_{3,3} + L = 4$ であるため、 $(3, 3)$ のベストな行き先が決まる。
- この場合、 $(3, 3)$ のベストな行き先は $(2, 4)$

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

例

- $(2, 2), (1, 4)$ の周りを見る。
- $(2, 4)$ と $(1, 4)$ がマージされる。
- $(3, 4)$ のベストな行き先は $(1, 4)$

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

例

- $(1, 2)$ の周りを見る。
- $(1, 1), (2, 2)$ がマージされる。
- $(1, 1)$ のベストな行き先は $(1, 2)$

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

例

- 色々マージされる。
- $(2, 4)$ のベストな行き先は $(2, 1)$
- $(2, 3), (3, 1)$ としても良い。

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

前計算

- $T_{i,j}$ の小さい順に、自分の周りで、自分の値以下であるものと merge するという操作をする。
- $T_{i,j}$ の値が $T_{a,b} + L$ 以下であるものに対する操作が全て終わったタイミングで、 $T_{a,b}$ が含まれる連結成分の中にある $T_{i,j}$ が最も大きい (i, j) を確認する。
- 時間計算量は $O(HW \log(HW))$ や $O(HW \alpha(HW))$ で求められる。

クエリ処理

「 (A_k, B_k) から x 回のハイジャンプで (C_k, D_k) に辿り着けるか」という判定問題は、以下のように置き換えれる。

- (A_k, B_k) から今いるマスのベストな行き先に $(x - 1)$ 回移動したときのマスを (a, b) としたときに、 (a, b) から 1 回のハイジャンプで (C_k, D_k) に辿り着けるか。

クエリ処理

「 (A_k, B_k) から x 回のハイジャンプで (C_k, D_k) に辿り着けるか」という判定問題は、以下のように置き換えれる。

- (A_k, B_k) から今いるマスのベストな行き先に $(x - 1)$ 回移動したときのマスを (a, b) としたときに、 (a, b) から 1 回のハイジャンプで (C_k, D_k) に辿り着けるか。

クエリ処理

- (A_k, B_k) から今いるマスのベストな行き先に $(x - 1)$ 回移動したときのマスを高速に求めたい。
- 事前にダブリングしておくことで、 $O(\log(HW))$ で求められる。
 - (a, b) からベストの行き先に x 回移動したときのマスを、 $x = 1, 2, 4, \dots, 2^t$ に対して求めておく。

再掲

Q. マス (a, b) から 1 回のハイジャンプで行けるマス (i, j) は？

A. 書かれている数が $T_{a,b} + L$ 以下の隣り合うマスの組を全て辺で結んだときに、 (a, b) と連結であるもの

クエリ処理

- 前計算と同じように、Union Find を用いると、全てのクエリに対する判定問題を $O(HW\alpha(HW))$ で答えられる。
- 並列二分探索すると、全体で $O((Q + HW \log(HW)) \log(HW))$ で求められる。
- x 回ベストな行き先に行くパートを工夫すれば時間計算量は $O((Q + HW \log(HW))\alpha(HW))$ になる。

小課題 5 : $H \times W \leq 300000, Q \leq 300000$

- 小課題 4 の方法でも定数倍によっては満点が取れます
- 前述したものより、クエリ処理の部分の計算量が良い解を紹介します。

クエリ処理

- 今まで同じ Union Find のマージを $O(\log(HW))$ 回してきた
 - 流石に無駄じゃない...?
- Union Find のマージ過程の木を考える。

クエリ処理

(A_k, B_k) と (C_k, D_k) がマージされたタイミングで見ていたマスに書かれていた値を X とする。

- これは Union Find のマージ過程を表す木の LCA を求める問題に帰着できて、適切な前計算の元でクエリごとに $O(\log(HW))$ で求められる。

以下の判定問題をもう一度考える。

- (A_k, B_k) から今いるマスのベストな行き先に $(x - 1)$ 回移動したときのマスを (a, b) としたときに、 (a, b) から 1 回のハイジャンプで (C_k, D_k) 辿り着けるか。

クエリ処理

- (A_k, B_k) から今いるマスのベストな行き先に $(x - 1)$ 回移動したときのマスを (a, b) としたときに、 (a, b) から 1 回のハイジャンプで (C_k, D_k) 辿り着けるか。

この判定問題は以下のように解ける。

- $X \leq T_{a,b} + L$ ならば可能、そうでないなら不可能。

よって、判定問題を $O(1)$ で解けるため、二分探索することでクエリごとに $O(\log(HW))$ で答えが求まる。

統計情報

点数	人数	累計人数
100	13	13
80	3	16
50	1	17
30	1	18
10	5	23
0	3	26

