

memset0's Notebook

方知蓦然回首之时
那人却已不在灯火阑珊处

关于我
友情链接
文章聚合

Theme Ringo by memset0
Proudly powered by Typecho

洛谷3350 [ZJOI2016]旅行者

2019-02-24 | 题解

给定一个网格图，边有边权，给定 M 个查询，求两点最短路。

考虑离线所有询问，然后分治，假设当前分治一条横穿中心的线，枚举每个线上节点更新当前所有询问的答案。

- 若某查询的两点在此线两侧，则最短路一定能在此次被更新
- 若某查询的两点在此线同侧，则最短路可能被当前更新，可以递归下去做。

可以证明每次选当前分治矩形的最长边，时间复杂度为 $O(n\sqrt{n}\log n)$ 。

代码：

```
// =====  
//   author: memset0  
//   date: 2019.02.24 11:31:22  
//   website: https://memset0.cn/  
// =====  
#include <bits/stdc++.h>  
#define ll long long  
namespace ringo {
```



memseto's Notebook

方知蓦然回首之时
那人却已不在灯火阑珊处

关于我
友情链接
文章聚合

Theme Ringo by memseto
Proudly powered by Typecho

```
template <class T> inline void read(T &x) {
    x = 0; register char c = getchar(); register bool f = 0;
    while (!isdigit(c)) f ^= c == '-', c = getchar();
    while (isdigit(c)) x = x * 10 + c - '0', c = getchar();
    if (f) x = -x;
}

template <class T> inline void print(T x) {
    if (x < 0) putchar('-'), x = -x;
    if (x > 9) print(x / 10);
    putchar('0' + x % 10);
}

template <class T> inline void print(T x, char c) { print(x), putchar(c); }

const int N = 1e5 + 10, M = 210, mov[4][2] = {{0, 1}, {1, 0}, {0, -1}, {-1, 0}};
int n, m, p, ans[N];
std::vector <std::vector <int> > map[4], dis;

struct node {
    int x, y;
} u, v;

struct status {
    node u; int w;
    inline bool operator < (const status &other) const {
        return w > other.w;
    }
} now; std::priority_queue <status> q;

struct info {
    int xl, xr, yl, yr, id;
}; std::vector <info> vet;

template <class T> void resize(T &a, int n, int m) {
    for (int k = 0; k < 4; k++) {
        a.resize(n + 1);
        for (int i = 0; i <= n; i++) a[i].resize(m + 1);
    }
}
```



memset0's Notebook

方知蓦然回首之时
那人却已不在灯火阑珊处

关于我
友情链接
文章聚合

Theme [Ringo](#) by [memset0](#)
Proudly powered by [Typecho](#)

```

}

template <class T> void memset(T &a, int w, int x1, int xr, int y1, int yr) {
    for (int i = x1; i <= xr; i++)
        for (int j = y1; j <= yr; j++)
            a[i][j] = w;
}

void spfa(int xs, int ys, int x1, int xr, int y1, int yr) {
    memset(dis, 1e9, x1, xr, y1, yr);
    q.push((status){(node){xs, ys}, 0}), dis[xs][ys] = 0;
    while (q.size()) {
        now = q.top(), u = now.u, q.pop(); if (now.w != dis[now.u.x][now.u.y])
        for (int t = 0, w; t < 4; t++) {
            v.x = u.x + mov[t][0], v.y = u.y + mov[t][1], w = map[t][u.x][u.y]
            if (v.x < x1 || v.x > xr || v.y < y1 || v.y > yr) continue;
            if (dis[u.x][u.y] + w < dis[v.x][v.y]) {
                dis[v.x][v.y] = dis[u.x][u.y] + w;
                q.push((status){v, dis[v.x][v.y]});
            }
        }
    }
}

void solve(int x1, int xr, int y1, int yr, std::vector <info> vet) {
    if (x1 > xr || y1 > yr || !vet.size()) return;
    std::vector <info> l_vet, r_vet;
    if ((xr - x1 + 1) * (yr - y1 + 1) < 100) {
        for (std::vector <info> ::iterator it = vet.begin(); it != vet.end(); it++)
            spfa(it->x1, it->y1, x1, xr, y1, yr);
        ans[it->id] = std::min(ans[it->id], dis[it->xr][it->yr]);
    } return;
}

if ((xr - x1 + 1) > (yr - y1 + 1)) {
    int mid = (x1 + xr) >> 1;
    for (int i = y1; i <= yr; i++) {
        spfa(mid, i, x1, xr, y1, yr);
    }
}

```



memset0's Notebook

方知蓦然回首之时
那人却已不在灯火阑珊处

关于我
友情链接
文章聚合

Theme [Ringo](#) by [memset0](#)
Proudly powered by [Typecho](#)

```

        for (std::vector<info>::iterator it = vet.begin(); it != vet.end();
              ans[it->id] = std::min(ans[it->id], dis[it->xl][it->yl] + dis
        }
        for (std::vector<info>::iterator it = vet.begin(); it != vet.end();
              if (it->xl < mid && it->xr < mid) l_vet.push_back(*it);
              if (it->xl > mid && it->xr > mid) r_vet.push_back(*it);
        }
        solve(xl, mid - 1, yl, yr, l_vet), l_vet.clear();
        solve(mid + 1, xr, yl, yr, r_vet), r_vet.clear();
    } else {
        int mid = (yl + yr) >> 1;
        for (int i = xl; i <= xr; i++) {
            spfa(i, mid, xl, xr, yl, yr);
            for (std::vector<info>::iterator it = vet.begin(); it != vet.end();
                  ans[it->id] = std::min(ans[it->id], dis[it->xl][it->yl] + dis
            }
            for (std::vector<info>::iterator it = vet.begin(); it != vet.end();
                  if (it->yl < mid && it->yr < mid) l_vet.push_back(*it);
                  if (it->yl > mid && it->yr > mid) r_vet.push_back(*it);
            }
            solve(xl, xr, yl, mid - 1, l_vet), l_vet.clear();
            solve(xl, xr, mid + 1, yr, r_vet), r_vet.clear();
        }
    }
}

void main() {
    read(n), read(m), resize(dis, n, m);
    for (int i = 0; i < 4; i++) resize(map[i], n, m);
    for (int i = 1; i <= n; i++)
        for (int j = 1; j < m; j++)
            read(map[0][i][j]), map[2][i][j + 1] = map[0][i][j];
    for (int i = 1; i < n; i++)
        for (int j = 1; j <= m; j++)
            read(map[1][i][j]), map[3][i + 1][j] = map[1][i][j];
    read(p);
    for (int i = 1; i <= p; i++) ans[i] = 1e9;
    for (int i = 1, xl, xr, yl, yr; i <= p; i++) {

```



memseto's Notebook

方知蓦然回首之时
那人却已不在灯火阑珊处

关于我
友情链接
文章聚合

Theme **Ringo** by **memseto**
Proudly powered by **Typecho**

洛谷3350 [ZJOI2016]旅行者 - memseto 的博客

```
read(xl), read(y1), read(xr), read(yr);  
vet.push_back((info){xl, xr, y1, yr, i});  
} solve(1, n, 1, m, vet);  
for (int i = 1; i <= p; i++) print(ans[i], '\n');  
}  
  
} signed main() { return ringo::main(), 0; }
```

巧妙的思路

分治

最短路

用户名

邮箱

网址 (选填)

可以在这里写评论哦 ~

提交评论



memset0's Notebook

方知蓦然回首之时
那人却已不在灯火阑珊处

MySQL 数据库基本操作学习笔记

[上一篇](#) «

UOJ 社区版安装笔记

» [下一篇](#)

© 2017 - 2019 memset0 的博客.

浙ICP备19006255号-1

97721 visits · 24757 visitors · 74.48 W words

在这里输入关键字哦 ~ (回车搜索)

[关于我](#)

[友情链接](#)

[文章聚合](#)

Theme [Ringo](#) by memset0

Proudly powered by [Typecho](#)

