dijkstra

人员

陈洛冉、赵熙羽、杨瑾硕、杨咏丞 到课

上周作业检查

上周作业链接: https://cppoj.kids123code.com/contest/1209

比赛概况 题目列表 选择题列表 提交记录 ★ 实时榜单 ★ 选择题排行榜

王向东老师周日十点半C++综合练习

刷新

#	用户名	姓名	编程分	时间	A	B	C	D	E
1	yangyongcheng	杨咏丞	400	18393	100	100	100		100
2	yuzijia1	于子珈	355	11840	100		85	70	100
3	xieyakai	谢亚锴	300	19358	100	100			100
4	zhaoxiyu	赵熙羽	200	18810	100	100			
5	niutongze	牛同泽	118	13943	100	18			
6	sunjingke	孙靖轲	100	6654	100				
7	yangjinshuo	杨谨硕	100	10064	100				
8	qinxiansen	秦显森	100	11373	100				
9	chenluoran	陈洛冉	100	11411	100				
10	siyunxin	司云心	100	13094	100				
11	liuchuangsu	刘闯速	100	14951	100				

本周作业

https://cppoj.kids123code.com/contest/1341 (课上讲了 A ~ C 题, 课后作业是 D 题)

课堂表现

今天讲了 dijkstra 这个算法, 这个算法思想不是很难, 但是要求同学们一定要把这个算法写熟, 课下需要多写几遍这个算法。

课堂内容

【模板】单源最短路径（弱化版）

n^2 的 dijkstra 算法

```cpp
#include <bits/stdc++.h>

using namespace std;

const int maxn = 1e4 + 5;
const int inf = (1LL<<31) - 1;
struct node {
```

```
  int to, value;
};
vector<node> vec[maxn];
int dis[maxn];
bool st[maxn];

int main()
{
  int n, m, s; cin >> n >> m >> s;
  while (m -- ) {
    int a, b, c; cin >> a >> b >> c;
    vec[a].push_back({b,c});
  }

  for (int i = 1; i <= n; ++i) dis[i] = inf;
  dis[s] = 0;

  for (int i = 0; i < n; ++i) {
    int min_dis = inf, id = -1;
    for (int j = 1; j <= n; ++j) {
      if (!st[j] && dis[j]<min_dis) min_dis = dis[j], id = j;
    }

    if (id == -1) break;

    st[id] = true;
    for (node it : vec[id]) dis[it.to] = min(dis[it.to], dis[id]+it.value);
  }

  for (int i = 1; i <= n; ++i) cout << dis[i] << " ";
  cout << endl;
  return 0;
}
```

【模板】单源最短路径（标准版）

堆优化的 dijkstra 模板题

```
#include <bits/stdc++.h>

using namespace std;

const int maxn = 1e5 + 5;
const int inf = 0x3f3f3f3f;
struct info {
  int to, value;
};
vector<info> vec[maxn];

struct node {
  int d, id;
```

```
  bool operator < (const node& p) const { return d < p.d; }
  bool operator > (const node& p) const { return d > p.d; }
};
int dis[maxn];
bool st[maxn];

int main()
{
  int n, m, s; cin >> n >> m >> s;
  while (m -- ) {
    int a, b, c; cin >> a >> b >> c;
    vec[a].push_back({b,c});
  }

  memset(dis, 0x3f, sizeof(dis));
  priority_queue<node, vector<node>, greater<node>> q;
  dis[s] = 0; q.push({dis[s],s});
  while (!q.empty()) {
    node u = q.top(); q.pop();
    int d = u.d, id = u.id;
    if (st[id]) continue;
    st[id] = true;

    for (info it : vec[id]) {
      if (!st[it.to] && d+it.value<dis[it.to]) {
        dis[it.to] = d+it.value; q.push({dis[it.to], it.to});
      }
    }
  }

  for (int i = 1; i <= n; ++i) cout << dis[i] << " ";
  cout << endl;
  return 0;
}
```

最小花费

把 dijkstra 过程中的累加过程换成累乘, 最小路程转化为最大乘积

```
#include <bits/stdc++.h>

using namespace std;

const int maxn = 2000 + 5;
struct info {
  int to; double value;
};
vector<info> vec[maxn];

struct node {
  double d; int id;
```

```
  bool operator < (const node& p) const { return d < p.d; }
  bool operator > (const node& p) const { return d > p.d; }
};
double dis[maxn];
bool st[maxn];

int main()
{
  int n, m; cin >> n >> m;
  while (m -- ) {
    int a, b, c; cin >> a >> b >> c;
    double t = (100.0-c) / 100.0;
    vec[a].push_back({b,t}); vec[b].push_back({a,t});
  }
  int A, B; cin >> A >> B;

  priority_queue<node, vector<node>, less<node>> q;
  dis[A] = 1; q.push({dis[A],A});
  while (!q.empty()) {
    node u = q.top(); q.pop();
    double d = u.d; int id = u.id;
    if (st[id]) continue;
    st[id] = true;

    for (info it : vec[id]) {
      if (!st[it.to] && d*it.value>dis[it.to]) {
        dis[it.to] = d*it.value; q.push({dis[it.to], it.to});
      }
    }
  }

  printf("%.8f\n", 100/dis[B]);
  return 0;
}
```