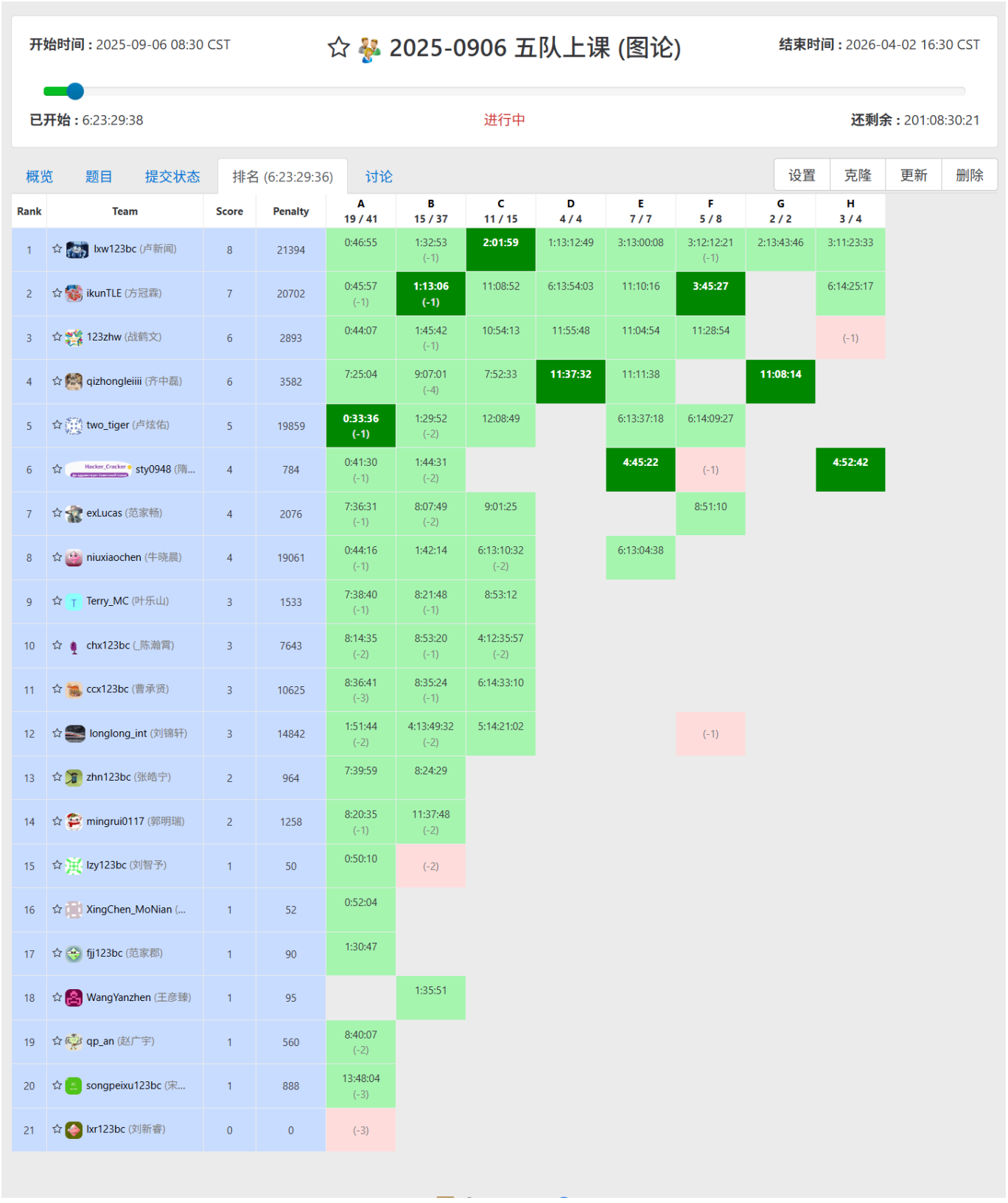
综合混练

人员

刘锦轩、牛晓晨、隋天翼、方冠霖、战鹤文、卢炫佑、刘智予、周游、王彦臻、宋沛旭、夏宇、史吉轩、蔡宇涵、卢新闻 到课

上周作业检查

上周作业链接: https://vjudge.net/contest/745738



开始时间 : 2025-09-06 08:30 CST

☆ 2025-0906 五队上课 (图论)

结束时间 : 2026-04-02 16:30 CST

已开始 : 6:23:29:38 进行中 还剩余 : 201:08:30:21

概览 题目 提交状态 排名 (6:23:29:36) 讨论 设置 克隆 更新 删除

Rank	Team	Score	Penalty	A 19 / 41	B 15 / 37	C 11 / 15	D 4 / 4	E 7 / 7	F 5 / 8	G 2 / 2	H 3 / 4
1	☆ lxw123bc (卢新闻)	8	21394	0:46:55	1:32:53 (-1)	**2:01:59**	1:13:12:49	3:13:00:08	3:12:12:21 (-1)	2:13:43:46	3:11:23:33
2	☆ ikunTLE (方冠霖)	7	20702	0:45:57 (-1)	**1:13:06 (-1)**	11:08:52	6:13:54:03	11:10:16	**3:45:27**		6:14:25:17
3	☆ 123zhw (战鹤文)	6	2893	0:44:07	1:45:42 (-1)	10:54:13	11:55:48	11:04:54	11:28:54		(-1)
4	☆ qizhongleiiii (齐中磊)	6	3582	7:25:04	9:07:01 (-4)	7:52:33	**11:37:32**	11:11:38		**11:08:14**	
5	☆ two_tiger (卢炫佑)	5	19859	**0:33:36 (-1)**	1:29:52 (-2)	12:08:49		6:13:37:18	6:14:09:27		
6	☆ sty0948 (隋...	4	784	0:41:30 (-1)	1:44:31 (-2)			**4:45:22**	(-1)		**4:52:42**
7	☆ exLucas (范家畅)	4	2076	7:36:31 (-1)	8:07:49 (-2)	9:01:25			8:51:10		
8	☆ niuxiaochen (牛晓晨)	4	19061	0:44:16 (-1)	1:42:14	6:13:10:32 (-2)		6:13:04:38			
9	☆ Terry_MC (叶乐山)	3	1533	7:38:40 (-1)	8:21:48 (-1)	8:53:12					
10	☆ chx123bc (_陈瀚霄)	3	7643	8:14:35 (-2)	8:53:20 (-1)	4:12:35:57 (-2)					
11	☆ ccx123bc (曹承贤)	3	10625	8:36:41 (-3)	8:35:24 (-1)	6:14:33:10					
12	☆ longlong_int (刘锦轩)	3	14842	1:51:44 (-2)	4:13:49:32 (-2)	5:14:21:02			(-1)		
13	☆ zhn123bc (张皓宁)	2	964	7:39:59	8:24:29						
14	☆ mingrui0117 (郭明瑞)	2	1258	8:20:35 (-1)	11:37:48 (-2)						
15	☆ lzy123bc (刘智予)	1	50	0:50:10	(-2)						
16	☆ XingChen_MoNian (...	1	52	0:52:04							
17	☆ fjj123bc (范家郡)	1	90	1:30:47							
18	☆ WangYanzhen (王彦臻)	1	95		1:35:51						
19	☆ qp_an (赵广宇)	1	560	8:40:07 (-2)							
20	☆ songpeixu123bc (宋...	1	888	13:48:04 (-3)							
21	☆ lxr123bc (刘新睿)	0	0	(-3)							

本周作业

https://vjudge.net/contest/747412, (课上讲了上周比赛的 D E F G H 和 prim 算法, 课后作业是本周比赛的 B C D E F 题)

课堂表现

同学们上课听讲整体都比较认真, 课上做题表现也都不错。

课堂内容

P5683 [CSP-J2019 江西] 道路拆除

从 A 点、s1 点、s2 点 分别跑 bfs, 然后枚举一个中间 i 点, 找 d[i]+d1[i]+d2[i] 的最小值

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

using namespace std;

const int maxn = 3000 + 5;
const int inf = 0x3f3f3f3f;
vector<int> vec[maxn];
int d[maxn], d1[maxn], d2[maxn];

void bfs(int id, int a[]) {
  for (int i = 0; i < maxn; ++i) a[i] = 0x3f3f3f3f;
  queue<int> q; q.push(id); a[id] = 0;
  while (!q.empty()) {
    int u = q.front(); q.pop();
    for (int i : vec[u]) {
      if (a[u]+1 < a[i]) {
        a[i] = a[u]+1; q.push(i);
      }
    }
  }
}

int main()
{
  int n, m; cin >> n >> m;
  for (int i = 1; i <= m; i++) {
    int u, v; cin >> u >> v;
    vec[u].push_back(v), vec[v].push_back(u);
  }

  int s1, t1, s2, t2; cin >> s1 >> t1 >> s2 >> t2;
  bfs(1, d), bfs(s1, d1), bfs(s2, d2);

  int res = inf;
  for (int i = 1; i <= n; ++i) {
    if (d[i]+d1[i]<=t1 && d[i]+d2[i]<=t2) {
      res = min(res, d[i]+d1[i]+d2[i]);
    }
  }
  if (res == inf) cout << -1 << endl;
  else cout << m - res << endl;
  return 0;
}
```

P2419 [USACO08JAN] Cow Contest S

把 (i,j) 的关系定义一个 f[i][j] 数组, 当 i 能战胜 j 时, 定义 f[i][j] 为 true

然后跑一遍 floyd, 枚举每个点 i, 如果 i 能唯一确定跟其他 n-1 个点的关系时, 它的排名就可以确定

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

using namespace std;

const int maxn = 100 + 5;
bool f[maxn][maxn];

int main()
{
  int n, m; cin >> n >> m;
  while (m -- ) {
    int a, b; cin >> a >> b; f[a][b] = true;
  }

  for (int k = 1; k <= n; ++k) {
    for (int i = 1; i <= n; ++i) {
      for (int j = 1; j <= n; ++j) {
        f[i][j] |= (f[i][k]&&f[k][j]);
      }
    }
  }

  int res = 0;
  for (int i = 1; i <= n; ++i) {
    int cnt = 0;
    for (int j = 1; j <= n; ++j) {
      if (j == i) continue;
      if (f[j][i] || f[i][j]) cnt++;
    }
    if (cnt == n-1) ++res;
  }
  cout << res << endl;
  return 0;
}
```

P5764 [CQOI2005] 新年好

以 1,a,b,c,d,e 这 6 个点为起点每个点跑一遍 bfs, 然后全排列 a,b,c,d,e 这 5 个点, 看怎么走最近

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

using namespace std;

typedef long long LL;
const int maxn = 50000 + 5;
const int inf = 0x3f3f3f3f;
```

```cpp
struct node {
  int to, value;
  bool operator > (const node& p) const { return value > p.value; }
};
vector<node> vec[maxn];
int a[10], dis[10][maxn];
bool st[maxn];

void bfs(int id) {
  memset(st, false, sizeof(st));
  priority_queue<node, vector<node>, greater<node>> q;
  q.push({a[id], 0}); dis[id][a[id]] = 0;

  while (!q.empty()) {
    node u = q.top(); q.pop();
    int u_id = u.to, u_dis = u.value;
    if (st[u_id]) continue;
    st[u_id] = true;

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

int main()
{
  int n, m; cin >> n >> m;
  map<int, int> mp;
  a[1] = 1; mp[1] = 1;
  for (int i = 2; i <= 6; ++i) cin >> a[i], mp[a[i]] = i;

  for (int i = 1; i <= m; ++i) {
    int u, v, w; cin >> u >> v >> w;
    vec[u].push_back({v,w}), vec[v].push_back({u,w});
  }

  memset(dis, 0x3f, sizeof(dis));
  for (int i = 1; i <= 6; ++i) bfs(i);

  LL res = 1e18;
  sort(a+2, a+6+1);
  do {
    LL sum = 0;
    for (int i = 2; i <= 6; ++i) sum += dis[mp[a[i-1]]][a[i]];
    res = min(res, sum);
  } while (next_permutation(a+2, a+6+1));
  cout << res << endl;
  return 0;
}
```

P3831 [SHOI2012] 回家的路

分层图, 把每个点拆分成两个点, 一个为横向的点, 一个为竖向的点

横向的点之间每行彼此建边, 竖向的点之间每列彼此建边, 同一个点的横线与竖向也建边

然后从起点往终点跑最短路即可

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

using namespace std;

const int maxn = 2e5 + 100;
const int inf = 0x3f3f3f3f;

int getId(int id, int c) { return id*2 + c; }
struct Point {
  int x, y;
} w[maxn];
struct Info {
  int id, c;
  bool operator < (const Info& p) const { return c < p.c; }
};
vector<Info> row[maxn], col[maxn];

struct node {
  int to, value;
  bool operator > (const node& p) const { return value > p.value; }
};
vector<node> vec[maxn];
int dis[maxn];
bool st[maxn];

int dijkstra(int _st, int _ed) {
  memset(dis, 0x3f, sizeof(dis));
  priority_queue<node, vector<node>, greater<node>> q; q.push({_st,0}); dis[_st] =
0;
  while (!q.empty()) {
    node u = q.top(); q.pop();
    int u_id = u.to, u_dis = u.value;
    if (st[u_id]) continue;
    st[u_id] = true;

    for (node it : vec[u_id]) {
      if (u_dis+it.value < dis[it.to]) {
        dis[it.to] = u_dis+it.value; q.push({it.to, dis[it.to]});
      }
    }
  }

  return (dis[_ed]==inf ? -1 : dis[_ed]);
}
```

```
int main()
{
  int n, m; cin >> n >> m;
  for (int i = 1; i <= m+2; ++i) {
    int x, y; cin >> x >> y; w[i] = {x, y};
    row[x].push_back({i,y}), col[y].push_back({i,x});
  }

  for (int i = 1; i <= n; ++i) {
    sort(row[i].begin(), row[i].end());
    int len = row[i].size();
    for (int j = 1; j < len; ++j) {
      Info a = row[i][j-1], b = row[i][j];
      int a_id = getId(a.id,1), b_id = getId(b.id,1);
      vec[a_id].push_back({b_id,(b.c-a.c)*2}), vec[b_id].push_back({a_id,(b.c-
a.c)*2});
    }
  }

  for (int j = 1; j <= n; ++j) {
    sort(col[j].begin(), col[j].end());
    int len = col[j].size();
    for (int i = 1; i < len; ++i) {
      Info a = col[j][i-1], b = col[j][i];
      int a_id = getId(a.id,2), b_id = getId(b.id,2);
      vec[a_id].push_back({b_id,(b.c-a.c)*2}), vec[b_id].push_back({a_id,(b.c-
a.c)*2});
    }
  }

  for (int i = 1; i <= m+2; ++i) {
    int id1 = getId(i,1), id2 = getId(i,2);
    vec[id1].push_back({id2,(i<=m)}), vec[id2].push_back({id1,(i<=m)});
  }

  cout << dijkstra(getId(m+1,1), getId(m+2,2)) << endl;
  return 0;
}
```

P1613 跑路

倍增 + floyd, 先求出来 i 到哪些 j 是 2^k 米, 也就是一秒能到达的

然后从点 1 跑 bfs 即可

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

using namespace std;

const int maxn = 50 + 5;
bool f[maxn][maxn][25];
```

```
set<int> s[maxn];
int dis[maxn];

void bfs() {
  memset(dis, -1, sizeof(dis));
  queue<int> q; q.push(1); dis[1] = 0;
  while (!q.empty()) {
    int u = q.front(); q.pop();
    for (int i : s[u]) {
      if (dis[i] == -1) { q.push(i), dis[i] = dis[u]+1; }
    }
  }
}

int main()
{
  int n, m; cin >> n >> m;
  while (m -- ) {
    int u, v; cin >> u >> v; f[u][v][0] = true;
  }
  for (int t = 1; t <= 20; ++t) {
    for (int k = 1; k <= n; ++k) {
      for (int i = 1; i <= n; ++i) {
        for (int j = 1; j <= n; ++j) {
          if (f[i][k][t-1] && f[k][j][t-1]) f[i][j][t] = true;
        }
      }
    }
  }

  for (int i = 1; i <= n; ++i) {
    for (int j = 1; j <= n; ++j) {
      for (int k = 0; k <= 20; ++k) {
        if (f[i][j][k]) s[i].insert(j);
      }
    }
  }

  bfs();
  cout << dis[n] << endl;
  return 0;
}
```

P3366 【模板】最小生成树

prim 模板

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

using namespace std;
```

```cpp
const int maxn = 5000 + 5;
const int inf = 0x3f3f3f3f;
int w[maxn][maxn], dis[maxn];
bool st[maxn];
int n, m;

int prim() {
    memset(dis, 0x3f, sizeof(dis));

    int res = 0;
    for (int i = 0; i < n; i++) {
        int id = -1;
        for (int j = 1; j <= n; ++j) {
            if (st[j]) continue;
            if (id==-1 || dis[j]<dis[id]) id = j;
        }

        if (i && dis[id]==inf) return -1;
        if (i) res += dis[id];

        st[id] = true;
        for (int j = 1; j <= n; j++) {
            if (!st[j]) dis[j] = min(dis[j], w[id][j]);
        }
    }
    return res;
}

int main()
{
    cin >> n >> m;
    memset(w, 0x3f, sizeof(w));
    while (m -- ) {
        int a, b, c; cin >> a >> b >> c;
        w[a][b] = w[b][a] = min(w[a][b], c);
    }

    int res = prim();

    if (res == -1) cout << "orz" << endl;
    else cout << res << endl;
    return 0;
}
```