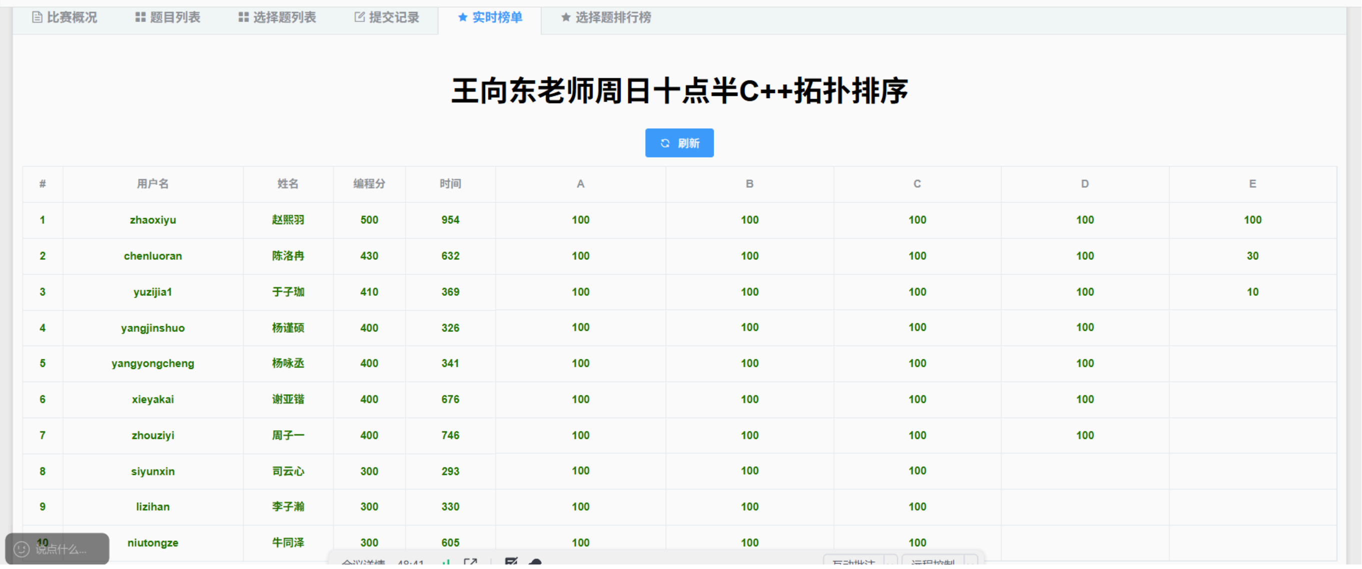
floyd

人员

赵熙羽、司云心、于子珈、陈洛冉、谢亚锴、杨咏丞、杨瑾硕、董浩桢、牟茗 到课, 周子一、孙靖轲 线上

上周作业检查

上周作业链接: https://cppoj.kids123code.com/contest/1688



王向东老师周日十点半C++拓扑排序

#	用户名	姓名	编程分	时间	A	B	C	D	E
1	zhaoxiyu	赵熙羽	500	954	100	100	100	100	100
2	chenluoran	陈洛冉	430	632	100	100	100	100	30
3	yuzijia1	于子珈	410	369	100	100	100	100	10
4	yangjinshuo	杨谨硕	400	326	100	100	100	100	
5	yangyongcheng	杨咏丞	400	341	100	100	100	100	
6	xieyakai	谢亚锴	400	676	100	100	100	100	
7	zhouziyi	周子一	400	746	100	100	100	100	
8	siyunxin	司云心	300	293	100	100	100		
9	lizihan	李子瀚	300	330	100	100	100		
10	niutongze	牛同泽	300	605	100	100	100		

本周作业

https://cppoj.kids123code.com/contest/1788 (课上讲了 A ~ D 题, 课后作业是 E 题, 其中 E 题正解会比较难一点, 同学们想不到正解的可以先打暴力, 老师下节课讲正解)

课堂表现

今天讲的 floyd 属于图论里面比较简单的算法, 同学们上课整体都吸收、掌握的比较好。

课堂内容

[NOIP 2013 普及组] 车站分级 (上周作业)

按照大小关系进行建图, 然后用 拓扑排序 跑最长路

建边过程中, 利用中间加一个 虚点, 可以减少建边的数量

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

using namespace std;

const int maxn = 2000 + 5;
struct node {
  int to, val;
};
vector<node> vec[maxn];
int w[maxn];
int din[maxn], f[maxn];

int main()
{
  int n, m; cin >> n >> m;
  for (int id = n+1; id <= n+m; ++id) {
    int len; cin >> len;
    for (int i = 1; i <= len; ++i) cin >> w[i];

    set<int> s1, s2;
    for (int i = w[1]; i <= w[len]; ++i) s2.insert(i);
    for (int i = 1; i <= len; ++i) s1.insert(w[i]), s2.erase(w[i]);

    for (int i : s2) vec[i].push_back({id,0}), ++din[id];
    for (int i : s1) vec[id].push_back({i,1}), ++din[i];
  }

  queue<int> q;
  for (int i = 1; i <= n+m; ++i) {
    if (!din[i]) q.push(i), f[i] = (i<=n);
  }
  while (!q.empty()) {
    int u = q.front(); q.pop();
    for (node it : vec[u]) {
      int to = it.to, val = it.val;
      --din[to]; f[to] = max(f[to], f[u]+val);
      if (!din[to]) q.push(to);
    }
  }

  int res = 0;
  for (int i = 1; i <= n; ++i) res = max(res, f[i]);
  cout << res << endl;
  return 0;
}
```

【模板】Floyd

floyd 模板, 求多源最短路

f[i][j]: i 到 j 的最短路是多少

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

using namespace std;

const int maxn = 100 + 5;
const int inf = 0x3f3f3f3f;
```

```
int f[maxn][maxn];

int main()
{
  int n, m; cin >> n >> m;
  memset(f, 0x3f, sizeof(f));
  for (int i = 1; i <= n; ++i) f[i][i] = 0;
  while (m -- ) {
    int a, b, c; cin >> a >> b >> c;
    f[a][b] = f[b][a] = min(f[a][b], c);
  }

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

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

[USACO07NOV] Cow Hurdles S

要求路径上最高的栏最小, 就在 floyd 的过程中, 把 dp 的方式改一下即可

改成 `f[i][j] = min(f[i][j], max(f[i][k],f[k][j]))` 即可

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

using namespace std;

const int maxn = 300 + 5;
const int inf = 0x3f3f3f3f;
int f[maxn][maxn];

int main()
{
  int n, m, T; cin >> n >> m >> T;

  memset(f, 0x3f, sizeof(f));
  for (int i = 1; i <= n; ++i) f[i][i] = 0;
  while (m -- ) {
    int a, b, c; cin >> a >> b >> c; f[a][b] = c;
  }

  for (int k = 1; k <= n; ++k) {
    for (int i = 1; i <= n; ++i) {
      for (int j = 1; j <= n; ++j) f[i][j] = min(f[i][j], max(f[i][k],f[k][j]));
    }
  }

  while (T -- ) {
    int a, b; cin >> a >> b;
    cout << (f[a][b]==inf ? -1 : f[a][b]) << endl;
  }
  return 0;
}
```

[USACO08JAN] Cow Contest S

f[i][j]: 代表 i 到 j 的大小是否已经知道了

跑 floyd, 然后明确所有点对的大小关系, 最后判断每个点跟其他点的关系是否都已知

```
#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;
}
```

灾后重建

某个村庄能用的话, 说明可以用这个村庄作为中转站跑 floyd 了

从前往后依次看每个时间点, 把这个时间点之前的村庄加入中转站跑 floyd 即可

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

using namespace std;

const int N = 200 + 5, M = 50000 + 5;
const int inf = 0x3f3f3f3f;
int w[N], f[N][N];

void update(int k, int n) {
  for (int i = 0; i < n; ++i) {
    for (int j = 0; j < n; ++j) f[i][j] = min(f[i][j], f[i][k]+f[k][j]);
  }
}

int main()
{
  int n, m; cin >> n >> m;
  for (int i = 0; i < n; ++i) cin >> w[i];

  memset(f, 0x3f, sizeof(f));
  for (int i = 0; i < n; ++i) f[i][i] = 0;
  while (m -- ) {
    int a, b, c; cin >> a >> b >> c;
    f[a][b] = f[b][a] = c;
  }

  int T, id = 0; cin >> T;
  while (T -- ) {
    int x, y, t; cin >> x >> y >> t;
    while (id<n && w[id]<=t) {
      update(id, n);
      ++id;
    }

    if (w[x]>t || w[y]>t || f[x][y]==inf) cout << -1 << endl;
    else cout << f[x][y] << endl;
  }
  return 0;
}
```