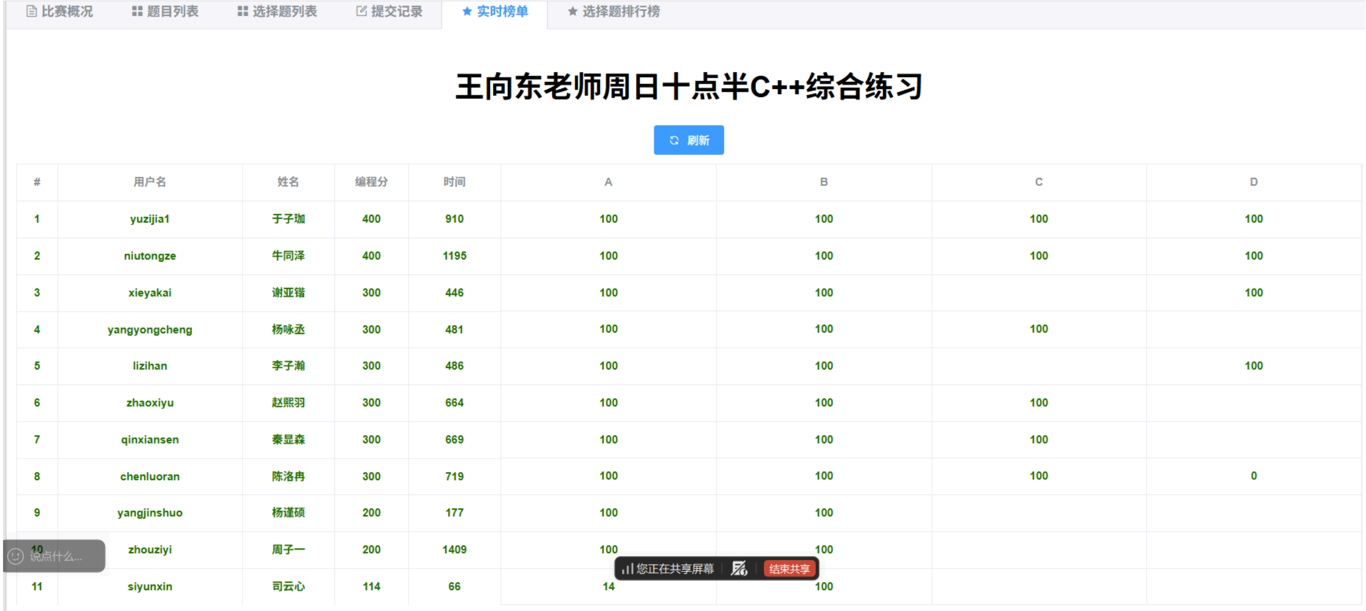
拓扑排序

人员

赵熙羽、司云心、牛同泽、于子珈、陈洛冉、谢亚锴、周子一、杨瑾硕 到课, 杨咏丞、李子瀚 线上

上周作业检查

上周作业链接: https://cppoj.kids123code.com/contest/1587



王向东老师周日十点半C++综合练习

#	用户名	姓名	编程分	时间	A	B	C	D
1	yuzijia1	于子珈	400	910	100	100	100	100
2	niutongze	牛同泽	400	1195	100	100	100	100
3	xieyakai	谢亚锴	300	446	100	100		100
4	yangyongcheng	杨咏丞	300	481	100	100	100	
5	lizihan	李子瀚	300	486	100	100		100
6	zhaoxiyu	赵熙羽	300	664	100	100	100	
7	qinxiansen	秦显森	300	669	100	100	100	
8	chenluoran	陈洛冉	300	719	100	100	100	0
9	yangjinshuo	杨谨硕	200	177	100	100		
10	zhouziyi	周子一	200	1409	100	100		
11	siyunxin	司云心	114	66	14	100		

本周作业

https://cppoj.kids123code.com/contest/1688 (课上讲了 A ~ D 题, 课后作业是 E 题, E 题同学们如果不好理解虚点的优化, 可以先写 O(n^3) 的做法)

课堂表现

今天讲的拓扑排序, 拓扑排序的内容整体是比较简单的, 大部分同学们课上也都掌握的比较好

今天的 C 题和 D 题是把 拓扑排序 和 dp 结合了一下, 这两道题需要同学们多自己画图理解一下。

课堂内容

[HAOI2006] 聪明的猴子 (上周作业)

先给原图中的点跑一遍最小生成树, 然后用最长的边, 跟每个猴子比即可

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

using namespace std;

const int M = 500 + 5, N = 1000 + 5;
```

```
int c[M];
struct node {
  int x, y;
} w[N];

struct Info {
  int a, b, value;
  bool operator < (const Info& p) const { return value < p.value; }
};
vector<Info> vec;
int dis(node a, node b) {
  int dx = a.x - b.x, dy = a.y - b.y;
  return dx*dx + dy*dy;
}

int f[N];
int fFind(int x) {
  if (f[x] != x) f[x] = fFind(f[x]);
  return f[x];
}

int main()
{
  int m; cin >> m;
  for (int i = 1; i <= m; ++i) cin >> c[i], c[i] *= c[i];
  int n; cin >> n;
  for (int i = 1; i <= n; ++i) cin >> w[i].x >> w[i].y;

  for (int i = 1; i <= n; ++i) {
    for (int j = i+1; j <= n; ++j) vec.push_back({i,j,dis(w[i],w[j])});
  }

  sort(vec.begin(), vec.end());
  for (int i = 1; i <= n; ++i) f[i] = i;
  int maxx = 0;
  for (Info it : vec) {
    int a = it.a, b = it.b, value = it.value;
    int fa = fFind(a), fb = fFind(b);
    if (fa != fb) f[fa] = fb, maxx = max(maxx, value);
  }

  int res = 0;
  for (int i = 1; i <= m; ++i) {
    if (c[i] >= maxx) ++res;
  }
  cout << res << endl;
  return 0;
}
```

【模板】拓扑排序 / 家谱树

拓扑排序模板题, 从入度为 0 的点入队, 然后往后搜, 遇到一个点就让他的入度 -1, 当一个点入度减为 0 时就把它入队

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

using namespace std;

const int maxn = 100 + 5;
vector<int> vec[maxn];
int din[maxn];

int main()
{
  int n; cin >> n;
  for (int i = 1; i <= n; ++i) {
    int x;
    while (true) {
      cin >> x;
      if (!x) break;
      vec[i].push_back(x); ++din[x];
    }
  }

  queue<int> q;
  for (int i = 1; i <= n; ++i) {
    if (!din[i]) q.push(i);
  }
  while (!q.empty()) {
    int u = q.front(); q.pop();
    cout << u << " ";
    for (int i : vec[u]) {
      --din[i];
      if (!din[i]) q.push(i);
    }
  }
  cout << endl;
  return 0;
}
```

[USACO17DEC] The Bovine Shuffle S

用拓扑排序判断有多少个点位于环上即可

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

using namespace std;

const int maxn = 1e5 + 5;
vector<int> vec[maxn];
int din[maxn];
int main()
{
  int n; cin >> n;
  for (int i = 1; i <= n; ++i) {
    int x; cin >> x; vec[i].push_back(x); ++din[x];
  }

  queue<int> q;
  for (int i = 1; i <= n; ++i) {
    if (!din[i]) q.push(i);
  }
  while (!q.empty()) {
    int u = q.front(); q.pop();
    --n;
    for (int i : vec[u]) {
      --din[i];
      if (!din[i]) q.push(i);
    }
  }

  cout << n << endl;
  return 0;
}
```

最大食物链计数

在拓扑排序的过程中维护 dp, f[i] 代表到 i 这个点时一共有多少条食物链

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

using namespace std;

const int maxn = 5e3 + 5;
const int mod = 80112002;
vector<int> vec[maxn];
int din[maxn], dout[maxn], f[maxn];

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

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

  int res = 0;
  for (int i = 1; i <= n; ++i) {
    if (!dout[i]) res = (res + f[i]) % mod;
  }
  cout << res << endl;
  return 0;
}
```

最长路

在拓扑的过程中做 dp, f[i] 用来存储以 i 结尾时, 最长路是多少

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

using namespace std;

const int maxn = 1500 + 5;
const int inf = 0x3f3f3f3f;
struct node {
  int to, val;
};
vector<node> vec[maxn];
int din[maxn], f[maxn];

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

  queue<int> q;
  for (int i = 1; i <= n; ++i) {
    f[i] = -inf;
    if (!din[i]) q.push(i);
  }
  f[1] = 0;
  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);
    }
  }

  cout << (f[n]==-inf ? -1 : f[n]) << endl;
  return 0;
}
```