对顶堆

人员

司云心、于子珈、谢亚锴 到课

上周作业检查

上周作业链接: https://cppoj.kids123code.com/contest/2684

王向东老师周日十点半C++折半搜索

刷新

#	用户名	姓名	编程分	时间	A	B	C
1	zhouziyi	周子一	300	5146	100	100	100
2	lizihan	李子瀚	300	5755	100	100	100
3	yuzijia1	于子珈	300	6764	100	100	100
4	chenluoran	陈洛冉	200	5304	100	100	
5	qinxiansen	秦显森	200	5591	100	100	
6	yangjinshuo	杨谨硕	200	6395	100	100	
7	xieyakai	谢亚锴	100	1534	100		
8	siyunxin	司云心	100	1594	100		

本周作业

https://cppoj.kids123code.com/contest/2901 (课上讲了 A ~ C 题, 课后作业是 D 题)

课堂表现

今天的 C 题会相对复杂一些, 需要在拓扑排序中维护 2 个 dp 信息, 需要同学们好好研究一下。

课堂内容

跑步 (上周作业)

n 很小, floyd 求最短路即可

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

using namespace std;

const int maxn = 20 + 5;
const int inf = 0x3f3f3f3f;
int dis[maxn][maxn];

int main()
{
  int n, m; cin >> n >> m;
  memset(dis, 0x3f, sizeof(dis));
```

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

  for (int i = 1; i <= n; ++i) {
    int a = i, b = i+1;
    if (i == n) b = 1;

    int x; cin >> x; dis[a][b] = dis[b][a] = x;
  }

  while (m -- ) {
    char s1[2], s2[2]; int x; cin >> s1 >> s2 >> x;
    int c1 = s1[0]-'A'+1, c2 = s2[0]-'A'+1;
    if (dis[c1][c2] == inf) dis[c1][c2] = dis[c2][c1] = x;
    else dis[c1][c2] = dis[c2][c1] = max(dis[c1][c2], x);
  }

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

  char s1[2], s2[2]; cin >> s1 >> s2;
  int c1 = s1[0]-'A'+1, c2 = s2[0]-'A'+1;
  cout << dis[c1][c2] << endl;
  return 0;
}
```

中位数

对顶堆, 用两个 优先队列 维护

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

using namespace std;

int main()
{
  priority_queue<int,vector<int>,less<int>> q1;
  priority_queue<int,vector<int>,greater<int>> q2;

  int n; cin >> n;
  for (int i = 1; i <= n; ++i) {
    int x; cin >> x;
    if (q1.empty() || x<=q1.top()) q1.push(x);
    else q2.push(x);

    while (q1.size() < q2.size()) {
      int u = q2.top(); q2.pop(); q1.push(u);
    }
```

```
    while (q2.size()+1 < q1.size()) {
      int u = q1.top(); q1.pop(); q2.push(u);
    }

    if (i&1) cout << q1.top() << endl;
  }
  return 0;
}
```

Least Elements

两个 multiset 维护, 用类似于对顶堆的思路做即可

```
#include <bits/stdc++.h>
#define int long long

using namespace std;

const int maxn = 2e5 + 5;
int w[maxn];

signed main()
{
  int n, m, K, sum = 0; cin >> n >> m >> K;
  for (int i = 1; i <= n; ++i) cin >> w[i];

  multiset<int> s1, s2;
  for (int i = 1; i <= n; ++i) {
    if (s1.empty() || w[i]<=*s1.rbegin()) s1.insert(w[i]), sum += w[i];
    else s2.insert(w[i]);

    if (i >= m) {
      while ((int)s1.size() < K) {
        int u = *s2.begin(); s2.erase(s2.find(u)); s1.insert(u); sum += u;
      }
      while ((int)s1.size() > K) {
        int u = *s1.rbegin(); s1.erase(s1.find(u)); sum -= u; s2.insert(u);
      }

      cout << sum << " ";

      int x = w[i-m+1];
      if (s2.count(x)) s2.erase(s2.find(x));
      else s1.erase(s1.find(x)), sum -= x;
    }
  }
  return 0;
}
```

游览

拓扑排序 + dp

f1[i]: 到 i 有多少条不同路径数量

f2[i]: 到 i 的不同路径的边权之和是多少

```
#include <bits/stdc++.h>
#define int long long

using namespace std;

const int maxn = 1e4 + 5;
const int mod = 10000;
struct node {
  int to, value;
};
vector<node> vec[maxn];
int deg[maxn], f1[maxn], f2[maxn];

signed main()
{
  int n, m, st, ed, tm; cin >> n >> m >> st >> ed >> tm;
  while (m -- ) {
    int a, b, c; cin >> a >> b >> c; vec[a].push_back({b,c});
    ++deg[b];
  }

  queue<int> q; q.push(st); f1[st] = 1;
  while (!q.empty()) {
    int u = q.front(); q.pop();
    for (node it : vec[u]) {
      int to = it.to, value = it.value;
      f1[to] = (f1[to] + f1[u]) % mod;
      f2[to] = (f2[to] + f2[u] + f1[u]*value) % mod;
      --deg[to];
      if (!deg[to]) q.push(to);
    }
  }

  cout << (f2[ed] + (f1[ed]-1)*tm) % mod << endl;
  return 0;
}
```