CSP 真题讲解2

人员

刘锦轩、牛晓晨、隋天翼、王彦臻、史吉轩、卢新闻、战鹤文、卢炫佑、咸俊永、彭贵晟、刘智予 到课

本周作业

https://www.luogu.com.cn/contest/282038 的 G H I J K L 题

https://www.luogu.com.cn/contest/282043 的 M N S 题

课堂表现

今天上课的题整体没有上节课难, 今天的题同学们都能补, 课下一定要把没做的题补一下。

课堂内容

P8817 [CSP-S 2022] 假期计划

只 O(n^2) 枚举中间的 B 点和 C 点, 前面的 A 点和 D 点可以预处理出来

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

using namespace std;

typedef long long LL;
const int maxn = 2500 + 5;
const int inf = 0x3f3f3f3f;
LL w[maxn];
vector<int> vec[maxn];
int f[maxn][maxn];

void bfs(int n, int id) {
  int dis[maxn]; memset(dis, 0x3f, sizeof(dis));
  queue<int> q; q.push(id); dis[id] = 0;
  while (!q.empty()) {
    int u = q.front(); q.pop();
    for (int i : vec[u]) {
      if (dis[i] == inf) q.push(i), dis[i] = dis[u]+1;
    }
  }

  for (int i = 1; i <= n; ++i) f[id][i] = dis[i];
}

struct node {
  int id; LL value;
  bool operator < (const node& p) const { return value < p.value; }
};
```

```
vector<node> vv[maxn];

bool check(int i, int j, int k, int l) {
  return (i!=j && i!=k && i!=l && j!=k && j!=l && k!=l);
}

int main()
{
  int n, m, K; cin >> n >> m >> K;
  for (int i = 2; i <= n; ++i) cin >> w[i];
  while (m -- ) {
    int a, b; cin >> a >> b;
    vec[a].push_back(b), vec[b].push_back(a);
  }

  for (int i = 1; i <= n; ++i) bfs(n, i);

  for (int i = 2; i <= n; ++i) {
    for (int j = 2; j <= n; ++j) {
      if (i==j || f[i][j]>K+1 || f[1][j]>K+1) continue;
      vv[i].push_back({j, w[j]});
    }
    sort(vv[i].begin(), vv[i].end()), reverse(vv[i].begin(), vv[i].end());
    while ((int)vv[i].size() > 3) vv[i].pop_back();
  }

  LL res = 0;
  for (int j = 2; j <= n; ++j) {
    for (int k = 2; k <= n; ++k) {
      if (j==k || f[j][k]>K+1) continue;
      for (node it1 : vv[j]) {
        for (node it2 : vv[k]) {
          int i = it1.id, l = it2.id;
          if (check(i,j,k,l)) res = max(res, w[i]+w[j]+w[k]+w[l]);
        }
      }
    }
  }
  cout << res << endl;
  return 0;
}
```

P8818 [CSP-S 2022] 策略游戏

只有 4 种情况

1. 最大的 a, 最小的 b
2. 最小的非负 a, 最小的 b
3. 最大的负 a, 最大的 b
4. 最小的 a, 最大的 b

4 种情况找最大的情况即可

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

using namespace std;

typedef long long LL;
const int N = 1e5 + 5, M = 20;
const int inf = 0x3f3f3f3f;
int a[N], b[N];
int fa[5][N][M], fb[3][N][M];
int _lg2[N];

int get_value(int f[][M], int l, int r, bool is_max) {
  int k = _lg2[r-l+1];
  int v1 = f[l][k], v2 = f[r-(1<<k)+1][k];
  return (is_max ? max(v1,v2) : min(v1,v2));
}

LL calc(int l1, int r1, int l2, int r2) {
  int a1 = get_value(fa[1], l1, r1, false), a2 = get_value(fa[2], l1, r1, true);
  int a3 = get_value(fa[3], l1, r1, false), a4 = get_value(fa[4], l1, r1, true);
  int b1 = get_value(fb[1], l2, r2, false), b2 = get_value(fb[2], l2, r2, true);

  LL v1 = -9e18, v2 = -9e18, v3 = -9e18, v4 = -9e18;
  if (a1 < 0) v1 = (LL)a1 * b2;
  if (a2 != -inf) v2 = (LL)a2 * b2;
  if (a3 != inf) v3 = (LL)a3 * b1;
  if (a4 >= 0) v4 = (LL)a4 * b1;

  return max({v1,v2,v3,v4});
}

int main()
{
  for (int i = 0; (1<<i) < N; ++i) _lg2[1<<i] = i;
  for (int i = 1; i < N; ++i) {
    if (!_lg2[i]) _lg2[i] = _lg2[i-1];
  }

  int n, m, T; cin >> n >> m >> T;
  for (int i = 1; i <= n; ++i) {
    cin >> a[i];
    for (int j = 1; j <= 4; ++j) fa[j][i][0] = a[i];
    if (fa[2][i][0] >= 0) fa[2][i][0] = -inf;
    if (fa[3][i][0] < 0) fa[3][i][0] = inf;
  }
  for (int i = 1; i <= m; ++i) {
    cin >> b[i];
    for (int j = 1; j <= 2; ++j) fb[j][i][0] = b[i];
  }

  for (int k = 1; k < M; ++k) {
```

```
    for (int i = 1; i+(1<<k)-1 <= n; ++i) {
      for (int j = 1; j <= 3; j += 2) {
        fa[j][i][k] = min(fa[j][i][k-1], fa[j][i+(1<<(k-1))][k-1]);
      }
      for (int j = 2; j <= 4; j += 2) {
        fa[j][i][k] = max(fa[j][i][k-1], fa[j][i+(1<<(k-1))][k-1]);
      }
    }
    for (int i = 1; i+(1<<k)-1 <= m; ++i) {
      fb[1][i][k] = min(fb[1][i][k-1], fb[1][i+(1<<(k-1))][k-1]);
      fb[2][i][k] = max(fb[2][i][k-1], fb[2][i+(1<<(k-1))][k-1]);
    }
  }

  while (T -- ) {
    int l1, r1, l2, r2; cin >> l1 >> r1 >> l2 >> r2;
//    cout << "------------------- ";
    cout << calc(l1, r1, l2, r2) << endl;
  }
  return 0;
}
```

P9752 [CSP-S 2023] 密码锁

从 0 ~ 99999 全枚举, 判断哪些情况合法

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

using namespace std;

const int maxn = 10 + 5;
int a[maxn][maxn], c[maxn];

bool check(int id) {
    int cnt = 0;
    for (int i = 1; i <= 5; ++i) {
        if (a[id][i] != c[i]) ++cnt;
    }

    if (cnt==0 || cnt>=3) return false;
    if (cnt == 1) return true;

    for (int i = 1; i <= 4; ++i) {
        if (a[id][i]!=c[i] && a[id][i+1]!=c[i+1]) {
            int x = c[i]-a[id][i], y = c[i+1]-a[id][i+1];
            if (x < 0) x += 10;
            if (y < 0) y += 10;
            if (x == y) return true;
        }
    }
    return false;
```

```
}

int main()
{
    int n; cin >> n;
    for (int i = 1; i <= n; ++i) {
        for (int j = 1; j <= 5; ++j) cin >> a[i][j];
    }

    int res = 0;
    for (int i = 0; i <= 99999; ++i) {
        c[1] = i/10000, c[2] = (i/1000)%10, c[3] = (i/100)%10;
        c[4] = (i/10)%10, c[5] = i%10;

        bool flag = true;
        for (int j = 1; j <= n; ++j) {
            if (!check(j)) flag = false;
        }
        if (flag) ++res;
    }
    cout << res << endl;
    return 0;
}
```

P9753 [CSP-S 2023] 消消乐

栈 + hash 维护

栈里面维护单个字符, 以及整个栈的 hash 值

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

using namespace std;

typedef long long LL;
typedef unsigned long long ULL;
const int maxn = 2e6 + 5;
const int P = 131;
char s[maxn];
struct node {
  char c;
  ULL val;
};

int main()
{
  int n; cin >> n >> (s+1);
  map<ULL, int> mp;
  ULL hVal = 1; ++mp[hVal];
  stack<node> stk; stk.push({' ', 1});
  for (int i = 1; i <= n; ++i) {
```

```
    if (!stk.empty() && s[i]==stk.top().c) stk.pop();
    else stk.push({s[i], stk.top().val*P+s[i]});

    ++mp[stk.top().val];
  }

  LL res = 0;
  for (auto it : mp) res += (LL)it.second*(it.second-1) / 2;
  cout << res << endl;
  return 0;
}
```

P11231 [CSP-S 2024] 决斗

一定是先用尽可能小的, 去打最小的

排序后再双指针跑一遍即可

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

using namespace std;

const int maxn = 1e5 + 5;
int w[maxn];

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

    int res = n;
    for (int i = 2, j = 1; i <= n; ++i) {
        if (w[i] > w[j]) --res, ++j;
    }
    cout << res << endl;
    return 0;
}
```

P11232 [CSP-S 2024] 超速检测

对于每辆车, 可以通过 二分 确定其超速的区间是哪一段

然后, 问题转化为了 有一些区间,想选最少的点覆盖这些区间

这就转化为了一个经典的贪心问题, 可以按照右端点排序然后贪心求

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

```cpp
using namespace std;

const int maxn = 1e5 + 5;
int n, m, L, V;
int d[maxn], v[maxn], a[maxn], p[maxn];
struct node {
  int x, y;
  bool operator < (const node& t) { return y < t.y; }
};
vector<node> vec;

void solve3(int x) {
  int pos = lower_bound(p+1, p+m+1, x) - p;
  if (pos <= m) vec.push_back({pos, m});
}

void solve2(int d, int v, int a) {
  if (a == 0) {
    if (v > V) solve3(d);
  }

  else if (a > 0) {
    if (v > V) solve3(d);
    else if (v == V) solve3(d+1);
    else { int len = (V*V - v*v) / (2*a); ++len; solve3(d+len); }
  }

  else {
    if (v > V) {
      int p1 = lower_bound(p+1, p+m+1, d) - p;
      if (p1 <= m) {
        int len = (V*V - v*v) / (2*a);
        if ((V*V - v*v) % (2*a) == 0) --len;
        int p2 = upper_bound(p+1, p+m+1, d+len) - p - 1;
        if (p1 <= p2) vec.push_back({p1, p2});
      }
    }
  }
}

void solve() {
  vec.clear();

  cin >> n >> m >> L >> V;
  for (int i = 1; i <= n; ++i) cin >> d[i] >> v[i] >> a[i];
  for (int i = 1; i <= m; ++i) cin >> p[i];
  for (int i = 1; i <= n; ++i) solve2(d[i], v[i], a[i]);
  cout << vec.size() << " ";

  sort(vec.begin(), vec.end());
  int last = -1e9, cnt = 0;
  for (node it : vec) {
    if (it.x > last) last = it.y, ++cnt;
```

```
  }
  cout << m - cnt << endl;
}

signed main()
{
  ios::sync_with_stdio(false);
  cin.tie(0);

  int T; cin >> T;
  while (T -- ) solve();
  return 0;
}
```

P5020 [NOIP 2018 提高组] 货币系统

先排序, 然后从小往大循环, 每次跑完全背包更新, 被标记的值不用选

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

using namespace std;

const int N = 100 + 5, M = 25000 + 5;
int w[N];
bool st[M];

void solve() {
  memset(st, false, sizeof(st)); st[0] = true;
  int n; cin >> n;
  for (int i = 1; i <= n; ++i) cin >> w[i];
  sort(w+1, w+n+1);

  int res = 0;
  for (int i = 1; i <= n; ++i) {
    int x = w[i];
    if (st[x]) continue;

    ++res;
    for (int j = x; j < M; ++j) st[j] |= st[j-x];
  }

//  cout << "------------------- ";
  cout << res << endl;
}

int main()
{
  int T; cin >> T;
  while (T -- ) solve();
```

```
  return 0;
}
```

P5022 [NOIP 2018 提高组] 旅行

树的话可以直接 dfs 遍历

基环树的话, 枚举删掉的边, 然后 dfs 遍历

```
#include <bits/stdc++.h>
#define x first
#define y second

using namespace std;

typedef pair<int, int> PII;
const int maxn = 5000 + 5;
vector<PII> edges;
set<int> s[maxn];
bool st[maxn];
int ans[maxn], temp[maxn];
int id;

void dfs(int u) {
  if (st[u]) return;
  st[u] = true, temp[++id] = u;
  for (int i : s[u]) dfs(i);
}

int main()
{
  int n, m; cin >> n >> m;
  for (int i = 1; i <= m; ++i) {
    int x, y; cin >> x >> y; edges.push_back({x,y});
    s[x].insert(y), s[y].insert(x);
  }

  if (m < n) {
    id = 0, dfs(1);
    for (int i = 1; i <= n; ++i) ans[i] = temp[i];
  }
  else {
    for (int i = 1; i <= n; ++i) ans[i] = n;
    for (PII it : edges) {
      int x = it.x, y = it.y;
      s[x].erase(y), s[y].erase(x);

      memset(st, false, sizeof(st));
      id = 0; dfs(1);
      if (id == n) {
        bool flag = false;
```

```
        for (int i = 1; i <= n; ++i) {
          if (temp[i] == ans[i]) continue;
          flag = (temp[i] < ans[i]);
          break;
        }
        if (flag) {
          for (int i = 1; i <= n; ++i) ans[i] = temp[i];
        }
      }

      s[x].insert(y), s[y].insert(x);
    }
  }
  for (int i = 1; i <= n; ++i) cout << ans[i] << " "; cout << endl;
  return 0;
}
```

P2679 [NOIP 2015 提高组] 子串

f[i][j][k][0/1]: 以 a 的第 i 位结尾, 匹配 b 的前 j 位, 分 k 段时, 第 i 个选/不选, 有多少方案

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

using namespace std;

typedef long long LL;
const int N = 1000 + 5, M = 200 + 5;
const int mod = 1e9 + 7;
char a[N], b[M];
int p[M][M][2], f[M][M][2];

int main()
{
  int n, m, k; cin >> n >> m >> k;
  cin >> (a+1) >> (b+1);

  p[0][0][0] = 1;
  for (int i = 1; i <= n; ++i) {
    memset(f, 0, sizeof(f));
    for (int j = 0; j <= m; ++j) {
      for (int k = 0; k <= m; ++k) {
        f[j][k][0] = (p[j][k][0] + p[j][k][1]) % mod;
        if (j>=1 && k>=1 && a[i]==b[j]) {
          f[j][k][1] = ((LL)p[j-1][k-1][0] + p[j-1][k-1][1] + p[j-1][k][1]) % mod;
        }
      }
    }
    memcpy(p, f, sizeof(p));
  }
  cout << (f[m][k][0] + f[m][k][1]) % mod << endl;
```

```
    return 0;
}
```