综合混练

人员

褚锦轩、许睿谦、王毅博、阮文璋、王承周、司云心 到课

上周作业检查

https://www.luogu.com.cn/contest/244893

2025-0504六队上课(综合混练)

报名 编辑比赛

题目数 4 | 报名人数 20

比赛说明 题目列表 排行榜

名次	参赛者	总分	A	B	C	D
#1	杨俊彦	400 (3.01h)	100 (615ms)	100 (837ms)	100 (124ms)	100 (3.01h)
#2	徐思远2	400 (5.15h)	100 (327ms)	100 (330ms)	100 (87ms)	100 (5.15h)
#3	白芸珲	400 (2.08d)	100 (370ms)	100 (323ms)	100 (13.24h)	100 (1.53d)
#4	王承周	400 (8.61d)	100 (667ms)	100 (330ms)	100 (2.43d)	100 (6.18d)
#5	刘奕辰	400 (9.04d)	100 (669ms)	100 (389ms)	100 (2.46d)	100 (6.57d)
#6	袁晨峻	400 (19.05d)	100 (6.32d)	100 (327ms)	100 (6.36d)	100 (6.38d)
#7	陈欣妙	300 (1.16s)	100 (695ms)	100 (387ms)	100 (81ms)	
#8	周治润	300 (1.34d)	100 (635ms)	100 (581ms)		100 (1.34d)
#9	褚锦轩	300 (7.10d)	100 (850ms)	100 (356ms)	100 (7.10d)	
#10	杨咏丞	300 (8.94d)	100 (3.41d)	100 (328ms)	100 (5.53d)	
#11	王毅博	284 (18.33d)	100 (5.58d)	100 (5.56d)		84 (7.20d)
#12	SSJ司云心	200 (948ms)	100 (592ms)	100 (356ms)	0	
#13	王博涵	0 (0ms)	0			

作业

https://www.luogu.com.cn/contest/246063 (课上讲了 A ~ C 题, 课后作业是 D 题)

课堂表现

今天的 A、B 2 道题, 同学们课上做的整体不是很好, 后来很多同学在老师的讲解下改出来了, 但是应该是没有完全掌握, 课下要再好好写一写这两道题。

课堂内容

P3131 [USACO16JAN] Subsequences Summing to Sevens S

把所有前缀和取余一下 7, 找 0 1 2 ... 6 每个前缀和对应的最前面和最后面的位置即可。

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

using namespace std;

typedef long long LL;
const int maxn = 5e4 + 5;
int w[maxn];
LL pre[maxn];
int p[10], s[10];

int main()
{
  int n; cin >> n;
  for (int i = 1; i <= n; ++i) cin >> w[i];
  for (int i = 1; i <= n; ++i) {
    pre[i] = pre[i-1] + w[i]; pre[i] %= 7;
  }

  for (int i = 0; i <= 6; ++i) p[i] = s[i] = -1;
  for (int i = 0; i <= n; ++i) {
    int t = pre[i];
    if (p[t] == -1) p[t] = i;
  }
  for (int i = n; i >= 0; --i) {
    int t = pre[i];
    if (s[t] == -1) s[t] = i;
  }

  int res = 0;
  for (int i = 0; i <= 6; ++i) {
    int l = p[i], r = s[i];
    if (l==-1 || r==-1) continue;
    res = max(res, r-l);
  }
  cout << res << endl;
  return 0;
}
```

P1025 [NOIP 2001 提高组] 数的划分

记忆化搜索, 对于 n 这个数, 分成 k 份, 最小值给 limit 开头时, 能有多少份。

此时 dfs(n,k,limit) 只需要调用 dfs(n-i,k-1,i) 即可 (i >= limit)

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

using namespace std;
```

```
int f[205][10][205];

int dfs(int n, int k, int limit) {
  if (n < limit) return 0;
  if (k == 1) return 1;
  if (f[n][k][limit]) return f[n][k][limit];

  int res = 0;
  for (int i = limit; i <= n; ++i) {
    res += dfs(n-i, k-1, i);
  }
  f[n][k][limit] = res;
  return f[n][k][limit];
}

int main()
{
  int n, k; cin >> n >> k;
  cout << dfs(n, k, 1) << endl;
  return 0;
}
```

P1330 封锁阳光大学

对每个联通块进行 dfs 黑白染色的搜索, 在每个联通块中找出黑、白块中少的那个, 然后全加起来。

黑白染色: 一个点染黑时, 周围点全部染白; 一个点染白时, 周围点全部染黑

如果中间出现了冲突, 则输出 -1

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

using namespace std;

const int maxn = 1e4 + 5;
vector<int> vec[maxn];
int f[maxn];
int cnt0, cnt1;

bool dfs(int u, int col) {
  if (f[u] != -1) return f[u]==col;

  f[u] = col;
  if (col == 0) cnt0++;
  else cnt1++;

  for (int i : vec[u]) {
    if (!dfs(i, 1-col)) return false;
  }
  return true;
}
```

```
int main()
{
  memset(f, -1, sizeof(f));
  int n, m; cin >> n >> m;
  while (m -- ) {
    int u, v; cin >> u >> v;
    vec[u].push_back(v), vec[v].push_back(u);
  }

  int res = 0;
  for (int i = 1; i <= n; ++i) {
    if (f[i] == -1) {
      cnt0 = 0, cnt1 = 0;
      if (!dfs(i, 0)) {
        cout << "Impossible" << endl;
        return 0;
      } else res += min(cnt0, cnt1);
    }
  }
  cout << res << endl;
  return 0;
}
```

U537158 floor

记忆化搜索, 维护 f[x][y][id] 的信息, 代表是否到过 (x,y) 这个点, 沿着 id 方向 这个状态

那么 dfs(x,y,id) 的下一个状态, 则就是看下一个位置 (x+dx[id],y+dy[id]) 是 '.' 还是 '#' 即可

如果是 '.', 下一个状态就是 dfs(x+dx[id], y+dy[id], id);

如果是 '#', 下四个状态就是 dfs(x, y, 0/1/2/3);

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

using namespace std;

const int maxn = 200 + 5;
char s[maxn][maxn];
bool f[maxn][maxn][4];
int dx[] = {0, 0, -1, 1}, dy[] = {-1, 1, 0, 0};
int n, m;

void dfs(int x, int y, int id) {
  if (f[x][y][id]) return;
  f[x][y][id] = true;

  int nx = x+dx[id], ny = y+dy[id];
  if (s[nx][ny] == '.') return dfs(nx, ny, id);

  for (int i = 0; i < 4; ++i) dfs(x, y, i);
```

```
}

int main()
{
  cin >> n >> m;
  for (int i = 1; i <= n; ++i) cin >> (s[i]+1);

  for (int i = 0; i < 4; ++i) dfs(2, 2, i);

  int res = 0;
  for (int i = 1; i <= n; ++i) {
    for (int j = 1; j <= m; ++j) {
      int cnt = 0;
      for (int k = 0; k < 4; ++k) {
        if (f[i][j][k] && s[i][j]=='.') cnt = 1;
      }
      res += cnt;
    }
  }
  cout << res << endl;
  return 0;
}
```