杂题混练

人员

刘佳赫、杨洋、刘子淇 到课

作业检查

上周作业链接：https://vjudge.net/contest/651049

密码：code@123

刘佳赫 完成 3 道题，杨洋 完成 1 道题，刘子淇 上周请假

作业

https://vjudge.net/contest/651049, 上周 3 道作业题要求大家补完

https://www.luogu.com.cn/contest/196356, 课上 D E 2道题要求大家补完

https://vjudge.net/contest/652462, 课后作业 A B C 3道题要求大家课后进行思考尝试

课堂表现

同学们课上听讲都很认真，也基本都听懂了，课下一定要花时间自己推一遍并且补题，这样收获才会更大。

课堂内容

CF1485D Multiples and Power Differences

构造, 先把所有数变为 1~16 的最小公倍数, 然后交错着把其中的一部分进行修改，从而满足条件3

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

using namespace std;

const int maxn = 500 + 5;
int a[maxn][maxn], b[maxn][maxn];

int main()
{
  int t = 1;
  for (int i = 1; i <= 16; ++i) t = t / __gcd(t, i) * i;

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

  for (int i = 1; i <= n; ++i) {
```

```
    for (int j = 1; j <= m; ++j) b[i][j] = t;
  }

  for (int i = 1; i <= n; ++i) {
    for (int j = ((i&1)?1:2); j <= m; j += 2) {
      b[i][j] -= a[i][j]*a[i][j]*a[i][j]*a[i][j];
    }
  }

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

CF1473D Program

维护前缀最小值和最大值，同时维护后缀最小值和后缀最大值

删除 l~r 后，最小值和最大值可能在前面、也可能在后面出现

下面以最小值举例：

在前面出现 -> preMin[l-1]

在后面出现 -> sufMin[r+1] - (p[r] - p[l-1]) -- 这里的 p[r] - p[l-1] 即为中间 l~r 这一段的变化量，因为删去中间这一段了，所以中间的变化量需要被减去。

然后 2 个综合取最小，即为最小值

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

using namespace std;

const int maxn = 2e5 + 5;
char s[maxn];
int pSum[maxn];
int p[2][maxn], suf[2][maxn];

int calc(int n, int l, int r) {
  int minn = p[0][l-1], maxx = p[1][l-1];
  if (r < n) {
    minn = min(minn, suf[0][r+1] - pSum[r] + pSum[l-1]);
    maxx = max(maxx, suf[1][r+1] - pSum[r] + pSum[l-1]);
  }
  return maxx - minn + 1;
}

void solve() {
  int n, m; cin >> n >> m;
  for (int i = 0; i <= n+2; ++i) pSum[i] = 0;
```

```
  for (int i = 0; i < 2; ++i) {
    for (int j = 0; j <= n+2; ++j) {
      p[i][j] = suf[i][j] = 0;
    }
  }

  cin >> (s+1);
  for (int i = 1; i <= n; ++i) {
    pSum[i] = pSum[i-1] + (s[i]=='-'?-1:1);
    p[0][i] = min(p[0][i-1], pSum[i]), p[1][i] = max(p[1][i-1], pSum[i]);
  }

  suf[0][n] = suf[1][n] = pSum[n];
  for (int i = n-1; i >= 1; --i) {
    suf[0][i] = min(suf[0][i+1], pSum[i]), suf[1][i] = max(suf[1][i+1], pSum[i]);
  }

  while (m -- ) {
    int l, r; cin >> l >> r;
//    cout << "-------------------- ";
    cout << calc(n, l, r) << endl;
  }
}

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

CF1286A Garland

f[i][j][0/1]: 在第 i 个位置, 有 j 个偶数, 结尾是 0(偶数)/1(奇数) 时, 对应的 f 值最小为多少

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

using namespace std;

const int maxn = 100 + 5;
int w[maxn], f[maxn][maxn][2];

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

  memset(f, 0x3f, sizeof(f));
  f[0][0][0] = f[0][0][1] = 0;

  for (int i = 1; i <= n; ++i) {
```

```
    for (int j = 0; j <= min(i, n/2); ++j) {
      if (j && w[i]%2==0) f[i][j][0] = min(f[i-1][j-1][0], f[i-1][j-1][1]+1);
      if (!w[i] || (w[i]&1)) f[i][j][1] = min(f[i-1][j][0]+1, f[i-1][j][1]);
    }
  }

  cout << min(f[n][n/2][0], f[n][n/2][1]) << endl;
  return 0;
}
```

T486221 number

类似于 bfs 的做法，用 queue 进行模拟即可

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

using namespace std;

typedef long long LL;

int main()
{
  int k; cin >> k;
  queue<LL> q;
  for (int i = 1; i <= 9; ++i) q.push(i);

  int cnt = 0;
  while (!q.empty()) {
    LL u = q.front(); q.pop();
    ++cnt;
    if (cnt == k) { cout << u << endl; break; }

    int ge = u%10;
    if (ge != 0) q.push(u*10 + ge-1);
    q.push(u*10 + ge);
    if (ge != 9) q.push(u*10 + ge+1);
  }
  return 0;
}
```

T485193 floor

用 bfs/dfs 做都可以，要额外维护一个方向信息

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

using namespace std;

const int maxn = 200 + 5;
```

```cpp
char s[maxn][maxn];
bool f[maxn][maxn][4];
int dx[] = {-1, 1, 0, 0}, dy[] = {0, 0, -1, 1};

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()
{
  int n, m; 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) {
      if (s[i][j] == '#') continue;
      bool flag = false;
      for (int k = 0; k < 4; ++k) flag |= f[i][j][k];
      res += flag;
    }
  }
  cout << res << endl;
  return 0;
}
```