综合混练

人员

杨咏丞、韩鸣蔚、刘奕辰、杨俊彦、龙沛轩、李锦澍、周治润、潘俊伊、袁晨峻、白芸珲、王博涵、王陆文龙、曹塬 到课, 李雨谦 线上

上周作业检查

https://www.luogu.com.cn/contest/238523

2025-0330六队上课(综合混练)

报名 编辑比赛

题目数 4 | 报名人数 24

比赛说明 题目列表 排行榜

名次	参赛者	总分	A	B	C	D
#1	袁晨峻	400 (3.21d)	100 (5.15h)	100 (242ms)	100 (1.49d)	100 (1.50d)
#2	杨俊彦	400 (3.52d)	100 (3.14h)	100 (2.02s)	100 (2.44h)	100 (3.28d)
#3	阮文璋	400 (4.52d)	100 (9.52h)	100 (1.74s)	100 (12.89h)	100 (3.58d)
#4	李锦澍	300 (1.86d)	100 (8.26h)	100 (1.67s)	100 (1.52d)	
#5	刘奕辰	300 (8.11d)	100 (1.54d)	100 (1.82s)	100 (6.57d)	
#6	王承周	300 (10.04d)	100 (2.44d)	100 (2.42d)		100 (5.17d)
#7	周治润	300 (10.22d)	100 (5.10d)	100 (1.58s)	100 (5.13d)	
#8	董昱含	300 (12.10d)	100 (5.38d)	100 (4.22h)	100 (6.55d)	
#9	徐思远	261 (14.38h)	40 (0ms)	100 (472ms)	21 (7.17h)	100 (7.20h)
#10	SSJ司云心	200 (3.57d)	100 (3.57d)	100 (1.72s)		
#11	王毅博	200 (4.79d)	100 (4.70h)	100 (4.60d)		
#12	陈欣妙	200 (6.00d)		100 (1.68s)	100 (6.00d)	0
#13	曹壈	200 (10.10d)	100 (6.54d)	100 (3.56d)		
#14	杨咏丞	200 (10.12d)	100 (5.54d)	100 (4.57d)	0 (0ms)	
#15	许睿谦	164 (1.06s)		100 (1.06s)	64 (0ms)	
#16	潘俊伊	110 (1.58s)	10 (0ms)	100 (1.58s)	0 (0ms)	
#17	韩鸣蔚	100 (1.66s)		100 (1.66s)		
#18	褚锦轩	100 (1.77s)		100 (1.77s)		
#19	李雨谦	100 (1.90h)	0 (0ms)	100 (1.90h)		
#20	龙沛轩	100 (2.33d)		100 (2.33d)		
#21	王陆文龙	77 (0ms)	0 (0ms)	0 (0ms)	77 (0ms)	
#22	王博涵	0 (0ms)		0 (0ms)		
#23	woaiwo	0 (0ms)		0 (0ms)		

作业

https://www.luogu.com.cn/contest/240400 (课上讲了 A ~ C 题, 课后作业是 D 题)

课堂表现

今天的 A 题、B 题 2 个题 思路、代码 都相对比较复杂一些, 王陆文龙、龙沛轩、杨俊彦、刘奕辰 这几位同学课上做的比较好, 提出表扬！！

课堂内容

P1215 [USACO1.4] 母亲的牛奶 Mother's Milk

3 维 dfs, 搜所有 a、b、c 桶内牛奶的情况即可

每次只有 6 种可能, a->b, a->c, b->a, b->c, c->a, c->b, 只有这 6 种倒牛奶的情况

从 (0,0,c) 的情况往后搜即可

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

using namespace std;

const int maxn = 20 + 5;
bool f[maxn][maxn][maxn];
vector<int> vec;
int A, B, C;

void dfs(int x, int y, int z) {
  if (f[x][y][z]) return;
  f[x][y][z] = true;
  if (x==0) vec.push_back(z);

  if (x+y <= B) dfs(0, x+y, z);
  else dfs(x-(B-y), B, z);

  if (x+z <= C) dfs(0, y, x+z);
  else dfs(x-(C-z), y, C);

  if (y+x <= A) dfs(y+x, 0, z);
  else dfs(A, y-(A-x), z);

  if (y+z <= C) dfs(x, 0, y+z);
  else dfs(x, y-(C-z), C);

  if (z+x <= A) dfs(z+x, y, 0);
  else dfs(A, y, z-(A-x));

  if (z+y <= B) dfs(x, z+y, 0);
  else dfs(x, B, z-(B-y));
}

int main()
{
  cin >> A >> B >> C;
  dfs(0, 0, C);
  sort(vec.begin(), vec.end());
  for (int i : vec) cout << i << " ";
  cout << endl;
  return 0;
}
```

P2390 地标访问

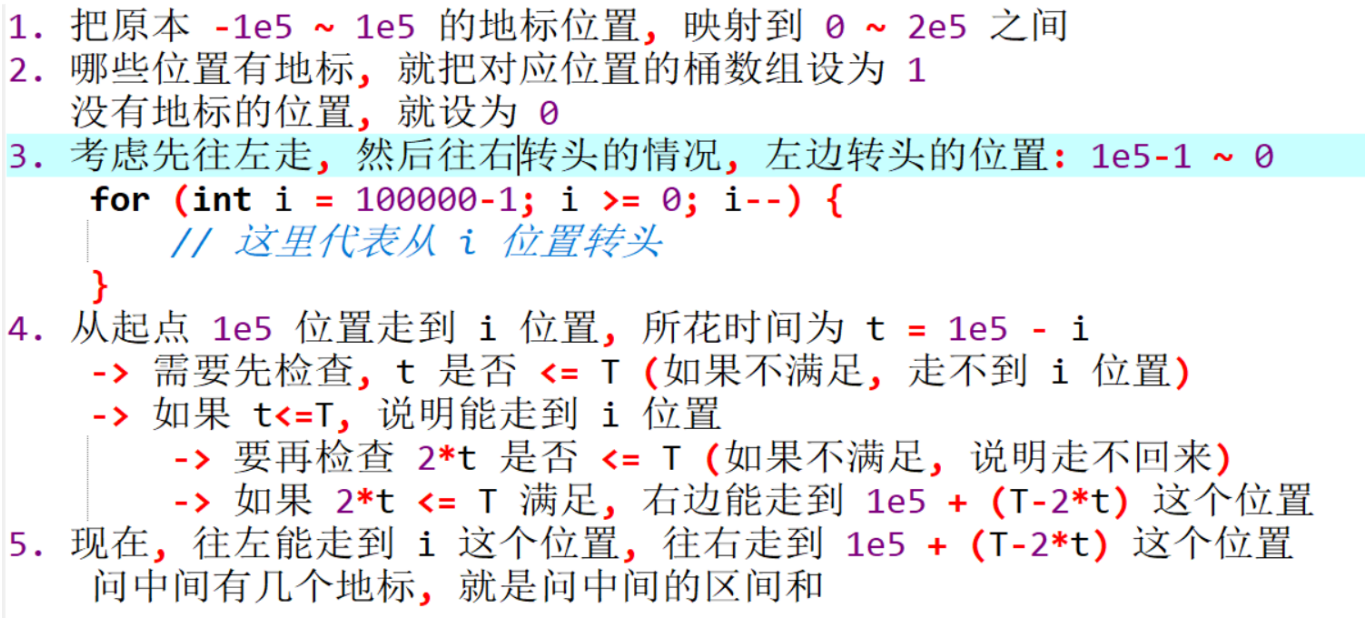
贪心思路: 一定是先往左走一段, 然后直接往右走; 或者是先往右走一段, 然后直接往左走

开一个 -1e5 ~ 1e5 的桶数组, 哪些位置有对应的地标, 就在对应桶的位置 +1 即可

然后枚举往左走一段的终点, 考虑在这个情况下往右最多走到哪; 再枚举往右走一段的终点, 考虑这种情况下往左最多走打哪。

中间经过多少点可以用前缀和来快速求

最后, 因为数组不能开负数, 给数组加一个偏移量即可, 把 -1e5 ~ 1e5 映射到 0 ~ 2e5即可




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

using namespace std;

const int N = 1e6 + 5;
int w[2*N], p[2*N];

int get_sum(int l, int r) { return p[r] - p[l-1]; }

int main()
{
  int n, m; cin >> m >> n;
  while (n -- ) { int x; cin >> x; w[x+N]++; }

  for (int i = 1; i < 2*N; ++i) p[i] = p[i-1] + w[i];

  int res = 0;
  for (int i = 0; i <= N; ++i) {
    if (N - i > m) continue;
    int j = max(N, min(m-2*(N-i)+N, 2*N-1));
    res = max(res, get_sum(i, j));
  }
```

```
  for (int i = 2*N-1; i >= N; --i) {
    if (i - N > m) continue;
    int j = min(N, max(N-m-2*(N-i), 0));
    res = max(res, get_sum(j, i));
  }
  cout << res << endl;
  return 0;
}
```

U548288 地标访问2

整体思路跟上个题类似, 也一定是先往左走一段, 然后直接往右走; 或者是先往右走一段, 然后直接往左走

不过数据范围不再是 -1e5 ~ 1e5, 而是 -1e9 ~ 1e9, 因此不能用桶来维护

也是先把所有位置放到一个数组中, 然后对数组中的这些位置排序

枚举往左走一段的终点, 考虑在这个情况下往右最多走到哪; 再枚举往右走一段的终点, 考虑这种情况下往左最多走打哪。

在这里考虑最右到哪或者最左到哪时, 可以用二分查找来进行优化即可

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

using namespace std;

const int maxn = 1e5 + 5;
const int inf = 0x3f3f3f3f;
int w[maxn];
int T, n, x = 0;

bool have_zero() {
  for (int i = 1; i <= n; ++i) {
    if (!w[i]) return true;
  }
  return false;
}

int get_zero_pos() {
  for (int i = 1; i <= n; ++i) {
    if (!w[i]) return i;
  }
  return 0;
}

int main()
{
    cin >> T >> n;
    for (int i = 1; i <= n; ++i) cin >> w[i];
    w[n+1] = -inf, w[n+2] = inf, n += 2;
    if (!have_zero()) w[n+1] = 0, n++, x = 1;
```

```
    sort(w+1, w+n+1);

    int pos = get_zero_pos(), res = 1;
    for (int i = pos-1; i >= 1; --i) {
    if (abs(w[i]) > T) break;
    res = max(res, pos-i+1);
    if (2*abs(w[i]) >= T) continue;
    int t = T - 2*abs(w[i]);
    int new_pos = upper_bound(w+1, w+n+1, t) - w - 1;
    res = max(res, new_pos-i+1);
    }

    for (int i = pos+1; i <= n; ++i) {
    if (w[i] > T) break;
    res = max(res, i-pos+1);
    if (2*w[i] >= T) continue;
    int t = T - 2*w[i];
    int new_pos = lower_bound(w+1, w+n+1, -t) - w;
    res = max(res, i-new_pos+1);
    }

    cout << res - x << endl;
  return 0;
}
```

P1367 蚂蚁

首先, 两只蚂蚁碰面后交换, 可以认为是没有发生交换, 那么 n 只蚂蚁最终的位置我们就可以确定了

然后, n 只蚂蚁的相对位置一定是不变的, 前面的还在前面, 后面的还在后面

这样处理一下, 就可以完成这个题了

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

using namespace std;

const int maxn = 1e5 + 5;
struct node {
  int pos, flag, id;
} a[maxn], b[maxn];

bool cmp(node p, node q) {
  if (p.pos != q.pos) return p.pos < q.pos;
  return p.flag < q.flag;
}

struct node2 {
  int pos, flag;
} ans[maxn];
```

```
int main()
{
  int n, t; cin >> n >> t;
  for (int i = 1; i <= n; ++i) {
    cin >> a[i].pos >> a[i].flag; a[i].id = i;
  }
  sort(a+1, a+n+1, cmp);

  for (int i = 1; i <= n; ++i) {
    b[i].flag = a[i].flag;
    if (a[i].flag == 1) b[i].pos = a[i].pos + t;
    else b[i].pos = a[i].pos - t;
  }
  sort(b+1, b+n+1, cmp);

  for (int i = 1; i <= n; ++i) {
    ans[a[i].id].pos = b[i].pos;
    if ((i>=2&&b[i].pos==b[i-1].pos) || (i<=n-1&&b[i].pos==b[i+1].pos))
ans[a[i].id].flag = 0;
    else ans[a[i].id].flag = b[i].flag;
  }

  for (int i = 1; i <= n; ++i) cout << ans[i].pos << " " << ans[i].flag << endl;
  return 0;
}
```