综合混练

人员

杨咏丞、李雨谦、韩鸣蔚、陈欣妙、刘奕辰、董昱含、杨俊彦、龙沛轩、李锦澍、周治润、潘俊伊、袁晨峻、徐思远、白芸珲、王博涵、曹塬、王陆文龙 到课

上周作业检查

https://www.luogu.com.cn/contest/237831

2025-0323六队上课(综合混练)

报名 编辑比赛 题目数 8 报名人数 23

比赛说明 题目列表 排行榜

名次	参赛者	总分	A	B	C	D	E	F	G	H
#1	杨俊彦	800 (15.64h)	100 (56.97min)	100 (57.13min)	100 (1.43h)	100 (1.67h)	100 (1.97h)	100 (2.41h)	100 (2.80h)	100 (3.46h)
#2	袁晨峻	800 (1.08d)	100 (39.72min)	100 (54.58min)	100 (1.08h)	100 (1.16h)	100 (1.73h)	100 (6.63h)	100 (6.47h)	100 (7.36h)
#3	阮文璋	800 (2.31d)	100 (5.44h)	100 (5.62h)	100 (5.86h)	100 (5.96h)	100 (6.82h)	100 (6.94h)	100 (9.01h)	100 (9.78h)
#4	刘奕辰	800 (8.30d)	100 (30.62min)	100 (37.78min)	100 (48.82min)	100 (51.52min)	100 (1.21h)	100 (1.63h)	100 (1.48d)	100 (6.58d)
#5	徐思远	744 (3.40d)	100 (46.30min)	100 (46.53min)	100 (1.83h)	100 (53.28min)	100 (1.71h)	44 (1.91h)	100 (1.53d)	100 (1.54d)
#6	白芸珲	700 (1.42d)	100 (36.35min)	100 (36.95min)	100 (55.88min)	100 (1.57h)	100 (1.28h)	100 (1.86h)		100 (1.13d)
#7	王承周	700 (9.80d)	100 (5.46h)	100 (5.68h)	100 (5.93h)	100 (6.04h)	100 (6.94h)	100 (2.43d)	100 (6.11d)	
#8	龙沛轩	700 (12.89d)	100 (58.90min)	100 (1.52h)	100 (1.24h)	100 (1.14h)	100 (1.73h)	100 (6.31d)	100 (6.31d)	
#9	褚锦轩	700 (17.69d)	100 (5.94h)	100 (5.93h)	100 (6.22h)	100 (6.82h)	100 (5.49d)	100 (5.55d)	100 (5.61d)	
#10	王毅博	677 (13.31d)	100 (5.95h)	100 (5.97h)	100 (6.33h)	100 (6.37h)	100 (6.81h)	77 (5.43d)		100 (6.57d)
#11	陈欣妙	633 (6.72d)	100 (30.58min)	100 (32.25min)	100 (42.15min)	100 (53.47min)	100 (1.71h)	33 (1.92h)	100 (6.46d)	
#12	董昱含	633 (13.52d)	100 (39.75min)	100 (1.06h)	100 (1.35h)	100 (1.42h)	100 (12.15h)	33 (6.56d)	100 (6.26d)	
#13	SSJ司云心	620 (17.40d)	100 (5.52h)	100 (6.06h)	100 (6.35h)	100 (6.42d)	100 (6.70h)		100 (3.50d)	20 (6.45d)
#14	李锦澍	600 (2.72d)	100 (38.17min)	100 (39.53min)	100 (51.98min)	100 (55.97min)	100 (1.75h)	100 (2.51d)		
#15	杨咏丞	600 (5.59d)	100 (49.52min)	100 (54.48min)	100 (1.82h)	100 (11.51h)	100 (1.46d)	100 (3.50d)		
#16	曹塬	600 (6.83d)	100 (47.92min)	100 (54.85min)	100 (1.57h)	100 (1.07h)	100 (1.89h)	100 (6.57d)		
#17	韩鸣蔚	590 (8.58h)	100 (42.12min)	100 (42.43min)	90 (1.92h)	100 (1.99h)	100 (1.40h)	100 (1.86h)		
#18	王博涵	580 (13.28d)	100 (41.80min)	100 (1.06h)	80 (1.42h)	100 (1.12h)	100 (6.55d)	100 (6.55d)		
#19	李雨谦	500 (4.90h)	100 (38.30min)	100 (43.53min)	100 (48.12min)	100 (1.00h)	100 (1.73h)			
#20	周治润	500 (5.20h)	100 (37.67min)	100 (38.43min)	100 (55.23min)	100 (1.11h)	100 (1.91h)			
#21	许睿谦	500 (1.25d)	100 (5.47h)	100 (5.49h)	100 (5.95h)	100 (6.23h)	100 (6.83h)			
#22	王陆文龙	500 (30.57d)	100 (6.09d)	100 (6.10d)	100 (6.12d)	100 (6.13d)	100 (6.14d)			
#23	潘俊伊	480 (5.52h)	100 (38.75min)	100 (55.47min)	80 (1.18h)	100 (50.68min)	100 (1.92h)			

作业

https://www.luogu.com.cn/contest/238523 (课上讲了 A ~ C 题, 课后作业是 D 题)

课堂表现

第一次用 OI 赛制上课, 同学们明显不适应, 也反映出同学们不能保证一次 AC 的准确度, 以后每次交题前要尽可能想全所有细节。

课堂内容

P1843 奶牛晒衣服

二分, 判断 mid 秒能否把所有衣服都烘干即可

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

using namespace std;

typedef long long LL;
const int maxn = 5e5 + 5;
int w[maxn];
int n, a, b;

int get_up(int x, int y) { return (x+y-1)/y; }

bool check(int mid) {
  int res = 0;
  for (int i = 1; i <= n; ++i) {
    if (a*mid >= w[i]) continue;
    res += get_up(w[i]-a*mid, b);
    if (res > mid) return false;
  }
  return true;
}

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

    int l = 1, r = 5e5 + 5;
    while (l <= r) {
    int mid = (l + r) / 2;
    if (check(mid)) r = mid-1;
    else l = mid+1;
    }
    cout << l << endl;
  return 0;
}
```

P2360 地下城主

bfs 板子题, 只是变成了 三维 问题

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

using namespace std;
```

```
const int maxn = 30 + 5;
char s[maxn][maxn][maxn];
int f[maxn][maxn][maxn];
struct node {
  int x, y, z;
};
int dx[] = {-1, 1, 0, 0, 0, 0};
int dy[] = {0, 0, -1, 1, 0, 0};
int dz[] = {0, 0, 0, 0, -1, 1};

int main()
{
    int n, m, c; cin >> n >> m >> c;
    int sx, sy, sz, ex, ey, ez;
    for (int i = 1; i <= n; ++i) {
    for (int j = 1; j <= m; ++j) {
      cin >> (s[i][j]+1);
      for (int k = 1; k <= c; ++k) {
        if (s[i][j][k] == 'S') sx=i,sy=j,sz=k;
        if (s[i][j][k] == 'E') ex=i,ey=j,ez=k;
      }
    }
    }

    memset(f, -1, sizeof(f));
    queue<node> q; q.push({sx,sy,sz}); f[sx][sy][sz] = 0;
    while (!q.empty()) {
    node u = q.front(); q.pop();
    int x = u.x, y = u.y, z = u.z;
    for (int i = 0; i < 6; ++i) {
      int nx = x+dx[i], ny = y+dy[i], nz = z+dz[i];
      if (nx>=1&&nx<=n&&ny>=1&&ny<=m&&nz>=1&&nz<=c&&s[nx][ny][nz]!='#'&&f[nx][ny]
[nz]==-1) {
        q.push({nx,ny,nz}); f[nx][ny][nz] = f[x][y][z]+1;
      }
    }
    }

    if (f[ex][ey][ez] == -1) cout << "Trapped!" << endl;
    else cout << "Escaped in " << f[ex][ey][ez] << " minute(s)." << endl;
  return 0;
}
```

P9241 [蓝桥杯 2023 省 B] 飞机降落

找一个最佳排列方式问题

N <= 10, 所以可以暴力枚举全部的排列, 看是否能有一个排列符合要求即可

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

```cpp
using namespace std;

const int maxn = 10 + 5;
int T[maxn], D[maxn], L[maxn];
int w[maxn];
int n;

bool check() {
  int last = -1;
  for (int i = 1; i <= n; ++i) {
    int id = w[i];
    int l = T[id], r = T[id] + D[id];
    if (last > r) return false;
    last = max(last, l) + L[id];
  }
  return true;
}

void solve() {
  cin >> n;
  for (int i = 1; i <= n; ++i) cin >> T[i] >> D[i] >> L[i];

  for (int i = 1; i <= n; ++i) w[i] = i;
  do {
    if (check()) { cout << "YES" << endl; return; }
  } while (next_permutation(w+1, w+n+1));
  cout << "NO" << endl;
}

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

P10417 [蓝桥杯 2023 国 A] 第 K 小的和

二分, 判断所有 ai+bj 的组合中, 是否有 >=k 个数满足 <=mid 的条件

check 的方法: 对每个 ai 来说, 看 b 数组中, 有多少满足 <= mid-ai 的数即可, 这里可以用 二分查找 加速

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

using namespace std;

typedef long long LL;
const int maxn = 1e5 + 5;
int a[maxn], b[maxn];
int n, m; LL k;
```

```
bool check(int mid) {
  LL res = 0;
  for (int i = 1; i <= n; ++i) {
    int x = mid - a[i];
    int pos = upper_bound(b+1, b+m+1, x) - b - 1;
    res += pos;
  }
  return res >= k;
}

int main()
{
  cin >> n >> m >> k;
  for (int i = 1; i <= n; ++i) cin >> a[i];
  for (int i = 1; i <= m; ++i) cin >> b[i];
  sort(b+1, b+m+1);

  LL l = 1, r = 2e9;
  while (l <= r) {
    int mid = (l + r) / 2;
    if (check(mid)) r = mid-1;
    else l = mid+1;
  }

  cout << l << endl;
  return 0;
}
```

P10416 [蓝桥杯 2023 国 A] XYZ

当 Z == 2*L 时, X/Y 的方案只有 1 种

当 Z == 2*L+1 时, X/Y 的方案有 2 种

当 Z == 2*L+2 时, X/Y 的方案有 3 种

...

当 Z == R 时, X/Y 的方案有 R-2*L+1 种

所以, 总答案应该是 1 + 2 + 3 + ... + R-2*L+1, 可以用等差数列 O(1) 求和

需要注意的问题: R-2*L+1 有可能 小于1, 此时说明无解, 应该输出 0

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

using namespace std;

typedef long long LL;

LL get_sum(int l, int r) {
  if (l > r) return 0;
```

```
  return ((LL)l+r)*(r-l+1)/2;
}

int main()
{
  int T; cin >> T;
  while (T -- ) {
    int l, r; cin >> l >> r;
    cout << get_sum(1, r-2*l+1) << endl;
  }
  return 0;
}
```

P1404 平均数

二分, 判断能否有一个长度 >=m 的区间的平均值满足 >=mid 的条件

check 方法: 每个数 -mid, 此时, 原问题转化为了 判断能否有一个长度 >=m 的区间的总和满足 >=0 的条件

此时, 可以用之前 最大子段和 的方法, 找到最大子段和, 看是否 >=0 即可

唯一区别是, 要满足长度 >=m 的要求, 所以求最大子段和时, 每次要用 p[i] - p_min[i-m]

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

using namespace std;

typedef long long LL;
const int maxn = 1e5 + 5;
int w[maxn];
int n, m;
int a[maxn];
LL p[maxn], p_min[maxn];

bool check(int mid) {
  for (int i = 1; i <= n; ++i) {
    a[i] = w[i] - mid, p[i] = p[i-1] + a[i];
    p_min[i] = min(p_min[i-1], p[i]);
  }

  for (int i = m; i <= n; ++i) {
    if (p[i] - p_min[i-m] >= 0) return true;
  }
  return false;
}

int main()
{
  cin >> n >> m;
  for (int i = 1; i <= n; ++i) cin >> w[i], w[i] *= 1000;

  int l = 0, r = 2000000;
```

```
    while (l <= r) {
      int mid = (l + r) / 2;
      if (check(mid)) l = mid+1;
      else r = mid-1;
    }
    cout << r << endl;
    return 0;
  }
```