

三分

人员

杨瑾硕、牛同泽、于霄龙、司云心 到课

上周作业检查

上周作业链接: <https://cppoj.kids123code.com/contest/967>

2025-1008 周日10:30 (综合练习)

刷新

#	用户名	姓名	编程分	时间	A	B	C	D
1	yangjinshuo	杨瑾硕	200	63	100	100		
2	sunjingke	孙靖柯	200	63	100	100		
3	gaojianhuan	高健桓	200	75	100	100		
4	zhaoxiyu	赵熙羽	200	77	100	100		
5	xieyakai	谢亚楷	200	78	100	100		
6	chenluoran	陈洛冉	200	80	100	100		
7	qinxiansen	秦显森	200	81	100	100	0	
8	yuzijia1	于子珈	200	103	100	100	0	
9	liuchuangsu	刘闯速	100	25	100			
10	yuxiaolong	于霄龙	100	28	100			
11	niutongze	牛同泽	0	0	0	0		

本周作业

<https://cppoj.kids123code.com/contest/1030> (课上讲了 A ~ C 题, 课后作业是 D 题)

课堂表现

今天给同学们重点讲了 三分 这个知识点, A、B 题要求同学们必须要熟练掌握, C 题可能比较难, 同学们可以课下好好复习复习这道题。

课堂内容

[CERC1998] 书的复制 (上周作业)

二分, 希望前面的人少抄写, 就是后面的人多抄写, 可以从后往前贪心, 让后面的人多抄写即可

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

using namespace std;

const int maxn = 500 + 5;
int w[maxn], ans1[maxn], ansr[maxn];
int m, k;
```

```

bool check(int mid) {
    int sum = 0, cnt = 1;
    ansr[cnt] = m;
    for (int i = m; i >= 1; --i) {
        if (sum + w[i] <= mid) sum += w[i];
        else ans1[cnt] = i+1, ++cnt, sum = w[i], ansr[cnt] = i;
    }
    ans1[cnt] = 1;
    return cnt <= k;
}

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

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

    check(1);
    for (int i = k; i >= 1; --i) cout << ans1[i] << " " << ansr[i] << endl;
    return 0;
}

```

【模板】三分 | 函数

三分 模板题

三分: 一般是针对单峰题用的

```

#include <bits/stdc++.h>

using namespace std;

const int maxn = 1e4 + 5;
const double eps = 1e-9;
int a[maxn], b[maxn], c[maxn];
int n;

double calc(double mid) {
    double res = -1e18;
    for (int i = 1; i <= n; ++i) {
        res = max(res, a[i]*mid*mid + b[i]*mid + c[i]);
    }
    return res;
}

```

```

void solve() {
    cin >> n;
    for (int i = 1; i <= n; ++i) cin >> a[i] >> b[i] >> c[i];

    double l = 0, r = 1000;
    while (r-l > eps) {
        double lmid = l + (r-l)/3, rmid = r - (r-l)/3;
        if (calc(lmid) < calc(rmid)) r = rmid;
        else l = lmid;
    }
    printf("%.4f\n", calc(l));
}

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

```

[CSP-J2022 山东] 宴会

把所有值全部乘 2, 方便后面避免小数运算, 最后输出结果的时候除 2 即可

很明显, 位置在中间某个位置的时候是最好的, 再最左最右都不好, 因此可以三分做

```

#include <bits/stdc++.h>

using namespace std;

const int maxn = 2e5 + 5;
struct node {
    int x, t;
    bool operator < (const node& p) const { return x < p.x; }
} w[maxn];
int n;

int calc(int mid) {
    int res = 0;
    for (int i = 1; i <= n; ++i) res = max(res, abs(w[i].x-mid)+w[i].t);
    return res;
}

void solve() {
    cin >> n;
    for (int i = 1; i <= n; ++i) cin >> w[i].x, w[i].x *= 2;
    for (int i = 1; i <= n; ++i) cin >> w[i].t, w[i].t *= 2;
    sort(w+1, w+n+1);

    int l = w[1].x, r = w[n].x;
    while (r - l >= 10) {

```

```

    int lmid = l + (r-l)/3, rmid = r - (r-l)/3;
    if (calc(lmid) < calc(rmid)) r = rmid;
    else l = lmid;
}

int id = l;
for (int i = l+1; i <= r; ++i) {
    if (calc(i) < calc(id)) id = i;
}

if (id&1) cout << id/2 << ".5" << endl;
else cout << id/2 << endl;
}

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

```

Sum

枚举以 i 结尾, 在前面的前缀和中找到 $\leq p[i]-k$ 的最大值

可以把前面的前缀和都放到一个 set 中, 在 set 中进行二分查找即可

```

#include <bits/stdc++.h>

using namespace std;

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

int main()
{
    LL n, k, P; cin >> n >> k >> P;
    for (int i = 1; i <= n; ++i) cin >> w[i], p[i] = (p[i-1] + w[i]) % P;

    set<LL> s; s.insert(0);
    LL res = 9e18;
    for (int i = 1; i <= n; ++i) {
        auto it = s.upper_bound((p[i]-k+P)%P); --it;
        LL val = (p[i]-*it+P)%P;
        if (val >= k) res = min(res, val);
        s.insert(p[i]);
    }
    cout << res << endl;
}

```

```
    return 0;  
}
```