三分

人员

柳力玮、李瑞涵、刘宸熙、王馨琪、初锦阳、韩昱辰、苑钊、宋云朗、温郝冬、蒋叔璋、田心一、赵牧之、纪博涵、刘派、高健桓、王恩泽 到课

上周作业检查

上周作业链接: https://cppoj.kids123code.com/contest/969

比赛概况 题目列表 选择题列表 提交记录 ★ 实时榜单 ★ 选择题排行榜

2025-1008 周日15:30 (综合练习)

刷新

#	用户名	姓名	编程分	时间	A	B	C	D
1	hanyuchen	韩昱辰	400	3703	100	100	100	100
2	liuliwei	柳力玮	400	7390	100	100	100	100
3	yuanzhao	苑钊	400	11148	100	100	100	100
4	liruihan	李瑞涵	300	3532	100	100		100
5	liuchenxi	刘宸熙	200	927	100	100		
6	chujinyang	初锦阳	200	1281	100	100		
7	wenhaodong	温郝冬	200	1337	100	100		
8	lizhishuo	李知朔	100	199	100	0		
9	wangxinqi	王馨琪	100	435	100			
10	zhaomuzhi	赵牧之	56	0	56			
11	tianxinyi	田心一	0	0		0		

本周作业

https://cppoj.kids123code.com/contest/1031 (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 题大部分同学课上没时间写这道题, 可以课下好好做做这道题。

课堂内容

驯鹿和雪橇 (上周作业)

先排序, 再维护前缀和, 最后在前缀和数组中做 二分查找

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

using namespace std;

typedef long long LL;
const int maxn = 2e5 + 5;
```

```
int w[maxn];
LL p[maxn];

int main()
{
  int n, T; cin >> n >> T;
  for (int i = 1; i <= n; ++i) cin >> w[i];
  sort(w+1, w+n+1);
  for (int i = 1; i <= n; ++i) p[i] = p[i-1] + w[i];

  while (T -- ) {
    LL x; cin >> x;
    int pos = upper_bound(p+1, p+n+1, x) - p - 1;
    cout << pos << 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;
}
```

ABA

枚举第 i 项作为中间的第二项, 分别看 i 前面和后面有多少 A/B/C/.../Z, 利用前缀和可以 O(1) 看

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

using namespace std;

typedef long long LL;
const int N = 2e5 + 5, M = 26 + 5;
char s[N];
int pre[N][M], suf[N][M];

int main()
{
  cin >> (s+1);
  int n = strlen(s+1);
  for (int i = 1; i <= n; ++i) {
    for (int j = 'A'; j <= 'Z'; ++j) pre[i][j-'A'] = pre[i-1][j-'A'] + (s[i]==j);
  }
  for (int i = n; i >= 1; --i) {
    for (int j = 'A'; j <= 'Z'; ++j) suf[i][j-'A'] = suf[i+1][j-'A'] + (s[i]==j);
  }

  LL res = 0;
  for (int i = 2; i <= n-1; ++i) {
    for (int j = 0; j < 26; ++j) res += (LL)pre[i-1][j]*suf[i+1][j];
  }
  cout << res << endl;
  return 0;
}
```