三分

人员

杨俊彦、曹塬、李锦澍、徐思远、隋天乙 到课

上周作业检查

上周作业链接: https://cppoj.kids123code.com/contest/1029

王向东老师周日八点半C++hash

刷新

#	用户名	姓名	编程分	时间	A	B	C	D
1	xusiyuan	徐思远	400	6399	100	100	100	100
2	yangjunyan	杨俊彦	400	8937	100	100	100	100
3	lijinshu	李锦澍	400	9384	100	100	100	100
4	yuanchenjun	袁晨峻	300	914	100	100	100	
5	chenxinmiao	陈欣妙	300	956	100	100	100	
6	yangyongcheng	杨咏丞	300	1163	100	100	100	
7	zhouzhirun	周治润	300	9442	100		100	100
8	liuyichen	刘奕辰	200	148	100		100	
9	liyuqian	李雨谦	100	90	100			

本周作业

https://cppoj.kids123code.com/contest/1114 (课上讲了 A ~ C 题, 课后作业是 D 题)

课堂表现

今天课上给同学们讲了三分, 同学们三分掌握的都很好, 但是有个严重的问题是都不认真读题, 这两道题目同学们第一次做题都没理解题目的正确意思, 以后要认真读题。

课堂内容

Tree and Hamilton Path 2

用 总边权*2 - 树的直径 即可

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

using namespace std;

typedef long long LL;
const int maxn = 2e5 + 5;
struct node {
  int to, val;
};
vector<node> vec[maxn];
```

```
LL d[maxn];

void dfs(int u, LL v) {
  if (d[u] != -1) return;

  d[u] = v;
  for (node it : vec[u]) dfs(it.to, v+it.val);
}

int main()
{
  int n; cin >> n;
  LL res = 0;
  for (int i = 1; i <= n-1; ++i) {
    int a, b, c; cin >> a >> b >> c; res += c*2;
    vec[a].push_back({b,c}), vec[b].push_back({a,c});
  }

  memset(d, -1, sizeof(d)); dfs(1, 0);
  int id = 1;
  for (int i = 2; i <= n; ++i) {
    if (d[i] > d[id]) id = i;
  }

  memset(d, -1, sizeof(d)); dfs(id, 0);
  LL maxx = 0;
  for (int i = 1; i <= n; ++i) maxx = max(maxx, d[i]);

  cout << res - maxx << 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;
}
```

Random Swaps of Balls

p1: 黑球保持原位不动的概率

p2: 黑球从其他位置来到 i 的概率

f[i][0]: i 轮之后黑球在位置 1 的概率

f[i][1]: i 轮之后黑球在其他单个位置的概率

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

using namespace std;

typedef long long LL;
const int maxn = 1e5 + 5;
const int mod = 998244353;
int f[maxn][2];

int qmod(int a, int k) {
  int res = 1;
  while (k) {
    if (k&1) res = (LL)res*a % mod;
    a = (LL)a*a % mod;
    k >>= 1;
  }
  return res;
```

```
}

int inv(int x) { return qmod(x, mod-2); }

int main()
{
  int n, k; cin >> n >> k;

  int p2 = 2 * inv((LL)n*n%mod) % mod;
  int p1 = (1 - (LL)p2*(n-1)%mod + mod) % mod;
  f[0][0] = 1, f[0][1] = 0;

  for (int i = 1; i <= k; ++i) {
    f[i][0] = ((LL)f[i-1][0]*p1%mod + ((LL)n-1)*f[i-1][1]%mod*p2%mod) % mod;
    f[i][1] = ((LL)f[i-1][0]*p2%mod + ((LL)n-2)*f[i-1][1]%mod*p2%mod + (LL)f[i-1]
[1]*p1%mod) % mod;
  }

  int res = (f[k][0] + (LL)f[k][1]*(n+2)%mod*(n-1)%mod*inv(2)%mod) % mod;
  cout << res << endl;
  return 0;
}
```