20190717作业&总结

今天myl学姐又来讲课啦，lqz可以放松一哈子了qwq
不过总觉得学姐的课堂气氛不活跃，lqz的课堂要活跃许多，许多在学姐课上特别沉默的孩纸在学长的课上却特别活跃，然后我和学姐说了说，她说了一句

> lqz比较懂他们，因为lqz就是那样的人

瞬间觉得好对qwq

不胡拉了，谈正事，~~（说的简直和fp一样）~~

今天学了分治、RMQ，上午的课基本都能听懂，但下午就辣鸡了，其实下午第一节课还是挺明白的，但是第二节课后面……我就基本上听不懂了(~~orz机房其他的大佬们~~)，没办法，菜是原罪嘛~

今日份代码

~~我可以说没有嘛？~~

P3865 【模板】ST表

模板题模板题~~~~~~~~不解释不解释
作为一个交作业的好孩纸，还是要搜一搜解释的啦

ST表类似树状数组，线段树这两种算法，是一种用于解决$RMQ(RangeMinimum/MaximumQuery$,即区间最值查询)问题的离线算法

与线段树相比，预处理复杂度同为$O(nlogn)$,查询时间上，ST表为$O(1)$,线段树为$O(nlogn)$

```
#include <iostream>
#include <cstdio>
#include <cstring>
#include <string>
#include <algorithm>
using namespace std;
```

```
const int N = 1000011;
int n, m, a[N], mx[N][30];

inline int read() {
    char c = getchar();
    int x = 0, f = 1;
    for( ; !isdigit(c); c = getchar()) if(c == '-') f = -1;
    for( ; isdigit(c); c = getchar()) x = (x << 3) + (x << 1) + (c ^ 48);
    return x * f;
}

inline void prepare() {
    for(int i = 1; i <= n; i++) mx[i][0] = a[i];
    for(int j = 1; (1 << j) <= n; j++) {
        for(int i = 1; i <= n; i++) {
            mx[i][j] = max(mx[i][j - 1], mx[i + (1 << (j - 1))][j - 1]);
        }
    }
}

inline int query(int l, int r) {
    int k = 1;
    while((1 << (k + 1)) <= r - l + 1) k++;
    return max(mx[l][k], mx[r - (1 << k) + 1][k]);
}

int main() {
    n = read(), m = read();
    for(int i = 1; i <= n; i++) a[i] = read();
    prepare();
    int l, r;
    for(int i = 1; i <= m; i++) {
        l = read(), r = read();
        cout << query(l, r) << '\n';
    }
}
```

P2571 [SCOI2010]传送带

三分套三分？手残打变量打到爆炸？疯了……只能抄题解

```
#include <iostream>
#include <cstdio>
#include <cmath>
using namespace std;
```

```
const double eps = 1e-8;
double ax, ay, bx, by, cx, cy, dx, dy, p, q, r;

double dis(double x1, double y1, double x2, double y2) {
    double xdis = x1 - x2, ydis = y1 - y2;
    return sqrt(xdis * xdis + ydis * ydis);
}

double f(double x1, double y1, double x2, double y2) {
    return dis(x1, y1, x2, y2)/r + dis(x2, y2, dx, dy)/q;
}

double calc1(double x, double y) {
    double lx = cx, ly = cy, rx = dx, ry = dy;
    while(dis(lx, ly, rx, ry) > eps) {
        double kl = (rx - lx) / 3.0, ky = (ry - ly) / 3.0;
        double lmidx = lx + kl, rmidx = rx - kl, lmidy = ly + ky, rmidy = ry
        double ans1 = f(x, y, lmidx, lmidy), ans2 = f(x, y, rmidx, rmidy);
        if(ans2 - ans1 > eps) rx = rmidx, ry = rmidy;
        else lx = lmidx, ly = lmidy;
    }
    return f(x, y, lx, ly);
}

double calc() {
    double lx = ax, ly = ay, rx = bx, ry = by;
    while(dis(lx, ly, rx, ry) > eps) {
        double kl = (rx - lx) / 3.0, ky = (ry - ly) / 3.0;
        double lmidx = lx + kl, rmidx = rx - kl, lmidy = ly + ky, rmidy = ry
        double ans1 = calc1(lmidx, lmidy) + dis(ax, ay, lmidx, lmidy)/p, ans2
        if(ans2 - ans1 > eps) rx = rmidx, ry = rmidy;
        else lx = lmidx, ly = lmidy;
    }
    return calc1(lx, ly) + dis(ax, ay, lx, ly)/p;
}

int main() {
    scanf("%lf%lf%lf%lf%lf%lf%lf%lf", &ax, &ay, & bx, &by, &cx, &cy, &dx, &dy
    scanf("%lf%lf%lf", &p, &q, &r);
    printf("%.2lf", calc());
}
```

# P3382 【模板】三分法

又是一个模板题，三分用于求单峰函数的极值，这里需要用到秦九韶公式，其他的就是个模板了

```
#include <iostream>
#include <cstdio>
#include <cstring>
#include <string>
using namespace std;

int n;
const double eps = 1e-6;
double l, r, a[20], mid, mmid, k;

double f(double x) {
    double s = 0;
    for(int i = 0; i <= n; i++) s = s * x + a[i];
    return s;
}

int main() {
    scanf("%d%lf%lf", &n, &l, &r);
    for(int i = 0; i <= n; i++) scanf("%lf", &a[i]);
    while(r - l >= eps) {
        k = (r - l) / 3.0;
        mid = l + k;
        mmid = r - k;
        if(f(mid) > f(mmid)) r = mmid;
        else l = mid;
    }
    printf("%.5lf\n", l);
    return 0;
}
```

CF1041C Coffee Break

贪心+二分，记录一下编号，排序一下，处理即可

```
#include <iostream>
#include <cstdio>
#include <algorithm>
using namespace std;

const int N = 1001111;
int n, k, m, d, b[N], p, pre, ans;
struct node {
    int id, wor, tag, bh;
} a[N];

inline int read() {
    char c = getchar();
    int x = 0, f = 1;
    for(; !isdigit(c); c = getchar()) if(c == '-') f = -1;
    for(; isdigit(c); c = getchar()) x = (x << 3) + (x << 1) + (c ^ 48);
    return x * f;
}

bool cmp(node a, node b) {
    return a.wor < b.wor;
}

int erbound(int now) {
    int l = now, r = n, mid, ans = n + 1;
    while(l <= r) {
        mid = (l + r) >> 1;
        if(a[mid].wor > a[now].wor + k) ans = mid, r = mid - 1;
        else l = mid + 1;
    }
    for(int i = ans; i <= n; i++) if(!a[i].tag) return i;
    return n + 1;
}

int main() {
    n = read(), m = read(), k = read();
    for(int i = 1; i <= n; i++) a[i].wor = read(), a[i].id = i;
    sort(a + 1, a + 1 + n, cmp);
    for(int i = 1; i <= n; i++) b[a[i].id] = i;
    p = 1;
    while(d != n) {
        ans++;
        for(int i = p; i <= n; i++) {
            if(!a[i].tag) {
                pre = i;
                p = pre + 1;
                break;
            }
        }
        a[pre].tag = 1;
        d++;
        a[pre].bh = ans;
        while(1) {
            int num = erbound(pre);
            if(num > n) break;
            a[num].bh = ans;
            a[num].tag = 1;
            d++;
            pre = num;
        }
    }
    cout << ans << '\n';
    for(int i = 1; i <= n; i++) {
        cout << a[b[i]].bh << ' ';
    }
    return 0;
}
```

UVA1152 和为0的4个数

如果直接暴力的话回是$O(n^4)$肯定跑不过去，所以先n^2处理a， b，然后二分寻找c， d，技巧

$$num = upper_bound(a,a+n,k) - lower_bound(a,a+n,k);$$

```
#include<bits/stdc++.h>
#define N 4005
#define LL long long
using namespace std;

int T,n,A[N],B[N],C[N],D[N],sum[N*N];

int main() {
    scanf("%d", &T);
    while(T--) {
        scanf("%d", &n);
        for(int i = 0; i < n; i++) {
            scanf("%d%d%d%d", &A[i], &B[i], &C[i], &D[i]);
        }
        int c = 0;
        for(int i = 0; i < n; i++) {
            for(int j = 0; j< n; j++) {
                sum[c++] = A[i] + B[j];
            }
        }
        stable_sort(sum, sum + c);
        LL ans=0;
        for(int i = 0; i < n; i++) {
            for(int j = 0; j < n; j++) {
                ans += upper_bound(sum, sum + c,-C[i]-D[j]) - lower_bound(su
            }
        }
        printf("%lld\n", ans);
        if(T) printf("\n");
    }
    return 0;
}
```