

2020/05/23-2020/05/30周报

团队训练

2020.5.27 [2018-2019 ACM-ICPC, Asia Jiaozuo Regional Contest](#)

李元恺

又摸了一周

姜维翰

专题

没有专题

比赛

没有比赛

题目

袁熙

专题

没有专题

比赛

没有比赛

题目

(暂放上周团队训练补题)

B.Tournament wqs二分, 单调队列

题意 x 轴上有 n 个点 $a_1, \dots, a_n$ 选择 x 轴上 k 个位置 $x_1, \dots, x_k$

使 $\sum_{i=1}^n \sum_{j=1}^k |a_i - x_j|$ 最小

数据范围 $n \leq 3 \times 10^5, a_i \leq 10^9$

思路: 直接dp是 $O(n^3)$ 的。对在选择数量上有约束的dp可以考虑wqs二分来降维; 再考虑dp的过程中, 具有决策单调性。对由 j 转移到的 i $\{ \text{forall } i' > i \}$ 对应的转移前的 $j' > j$ 一定是向更后面选位置 x_i 双向队列维护前述的性质dp就可以 $O(n \log^2 n)$ 做

代码

```
#include <bits/stdc++.h>
#define ll long long
#define tmp(x) std::cout<<"& "<<(x)<<" &\n"
#define rep(i,a,b) for(int i=(a);i<=(b);++i)
#define per(i,a,b) for(int i=(a);i>=(b);--i)
using namespace std;

const int maxn=3e5+100;
const int mo=998244353;
int n,k,st,ed;
int a[maxn],q[maxn],pos[maxn],cnt[maxn],tmp;
ll w[maxn],s[maxn];
inline ll getv(int i,int j){return s[i]+s[j]-s[i+j>>1]-s[i+j+1>>1];}
inline bool cmp(int a,int b,int c){
    ll al=w[a]+getv(a,c),bl=w[b]+getv(b,c);
    if(al==bl) return cnt[a]<cnt[b];
    return al<bl;
}
int work(ll mid){
    st=ed=1,w[0]=0;
    q[1]=0,pos[1]=1;
    rep(i,1,n){
        while(st<ed&&pos[st+1]<=i)++st;
        w[i]=w[q[st]]+getv(i,q[st])+mid;
        cnt[i]=cnt[q[st]]+1;
        while(st<=ed&&i<pos[ed]&&cmp(i,q[ed],pos[ed]))--ed;

        int l=pos[ed],r=n+1;
        while(l<r){
            int m=l+r>>1;
            if(cmp(i,q[ed],m))r=m;
            else l=m+1;
        }
        if(l<=n)q[++ed]=i,pos[ed]=l;
    }
}
```

```

        return cnt[n];
    }
    int main(){
        // freopen("in.txt", "r", stdin);
        scanf("%d%d", &n, &k);
        rep(i, 1, n) scanf("%d", &a[i]), s[i] = s[i-1] + a[i];
        ll l = 0, r = s[n] + 1;
        while(l + 1 < r){
            ll mid = l + r >> 1;
            tmp = work(mid);
            if(tmp >= k) l = mid + 1;
            else r = mid;
        }
        if(work(l) <= k) printf("%lld", w[n] - k * l);
        else if(work(l + 1) <= k) printf("%lld", w[n] - k * (l + 1));

        return 0;
    }

```

D.Country Meow 三分

题意：三维空间有100个点，求他们的最小球覆盖（最小化到这些点的距离最大值）

思路：最小圆覆盖模板/三分

三分：对x,y,z轴之一，所求值关于因变量是一个单峰函数。三分找到关于各个轴的最优位置即可。

代码

```

#include <bits/stdc++.h>
using namespace std;

const int maxn = 1000000;
struct point{
    double d[3];
}p[110], pl, pr, m1, m2;
int n;
inline double dis(point a, point b){
    return sqrt((a.d[0] - b.d[0]) * (a.d[0] - b.d[0]) + (a.d[1] - b.d[1]) * (a.d[1] - b.d[1]) + (a.d[2] - b.d[2]) * (a.d[2] - b.d[2]));
}
double work(int pos, double v){
    double l = -1e6, r = 1e6, ans = 1e9;
    if(pos) m1.d[pos - 1] = m2.d[pos - 1] = v;
    while(abs(l - r) > 1e-3){
        double r1 = 0, r2 = 0;
        if(pos != 2) r1 = work(pos + 1, (2 * l + r) / 3), r2 = work(pos + 1, (l + r * 2) / 3);
        else{
            m1.d[pos] = (2 * l + r) / 3, m2.d[pos] = (l + 2 * r) / 3;
            for(int
i = 1; i <= n; ++i) r1 = max(r1, dis(p[i], m1)), r2 = max(r2, dis(p[i], m2));
        }
        (r1 > r2) ? l = (2 * l + r) / 3 : r = (l + r * 2) / 3;
    }
}

```

```
        ans=min(r1,r2);
    }

    return ans;
}

int main(){
//    freopen("in.txt","r",stdin);
    scanf("%d",&n);
    for(int i=1;i<=n;++i){
        scanf("%lf%lf%lf",&p[i].d[0],&p[i].d[1],&p[i].d[2]);
    }
    printf("%.9lf\n",work(0,0));
    return 0;
}
```

本周推荐

姜维翰

袁熙

推荐三分

From:
<https://wiki.cvbbacm.com/> - CVBB ACM Team

Permanent link:
https://wiki.cvbbacm.com/doku.php?id=2020-2021:teams:acm_life_from_zero:5.23-5.30

Last update: 2020/06/06 14:28

