

开车旅行

一. 考察内容:

图论 倍增 树上倍增

二. 题目分析:

[题目大意]

给出一个城市序列，两个城市之间距离为这两个城市的高度差，AB交替开车，A喜欢开次短路，B喜欢开最短路，求出总旅行路程不超过X的情况下，如何选择起点，使得A开的路程/B开的路程最小，并对于一些询问，求出以S为起点，旅行路程不超过X时每个人开车的路程。

[写题思路]

首先把样例的图画出来，会发现A和B单独走的都是一棵树，于是我们考虑到通过求解树上问题解决该题，首先，求出对于每一个点，他的后继A点和后继B点分别是谁，考虑用双向链表实现，首先将所有城市按照高度正序排序，并用链表串起来，然后找到第一个城市，该城市左边没有城市，所以当前距离最短的城市一定是该城市排序后的左、右城市之一，距离次短的城市一定是该城市排序后左、右、左左、右右之一，求完该城市之后，我们更新链表，删掉第一个城市，求第二个城市，这时由于第一个城市已经被删掉了，所以剩下的城市中，第二个城市仍为第一个，按照上述方法重复求解即可。

根据上面求得的信息建树，分别对于A和B的路径建两棵树，（可以看成两棵节点相互重叠的树）对于这两棵树，我们把AB各走一次看做一步，维护 $f[i][j]$ 为节点i走 2^j 步之后所在的位置， $d[i][j]$ 为节点i到节点 2^j 之间需要走的距离， $dA[i][j]$ 为 $d[i][j]$ 中A走的部分， $dB[i][j]$ 为 $d[i][j]$ 中B走的部分，这样对于每个节点都可以logn求得以该点为起点，走不超过X距离时A、B分别走的长度，依照这个信息，可以很方便的求出题目中要求的信息。

三. 代码实现:

```
#define _CRT_SECURE_NO_DEPRECATED
/*****
*创建时间: 2018 09 16
*文件类型: 源代码文件
*题目来源: 洛谷
*当前状态: 已通过
*备忘录: 树上操作 倍增 双向链表
*作者: HtBest
*****/
#include <stdio.h>
#include <iostream>
#include <string>
#include <stdlib.h>
#include <string.h>
#include <algorithm>
#include <queue>
#include <bitset>
// #include <sys/wait.h>
// #include <sys/types.h>
// #include <unistd.h>
using namespace std;
#define MAXN 100001
#define lowbit(X) (X&-X)
int n,m,x,s,son1[MAXN],bro1[MAXN],fa1[MAXN],son2[MAXN],bro2[MAXN],fa2[MAXN],a[MAXN],
f[MAXN][20],dA[MAXN][20],dB[MAXN][20],d[MAXN][20];
struct C
{
```

```

    int id,h;
    bool operator < (C x) const
    {
        return h<x.h;
    }
};
/* Variable explain:
*/
void addt(int a,int b,int fa[],int son[],int bro[])
{
    bro[b]=son[a];
    son[a]=b;
    fa[b]=a;
}
void read()
{
    scanf("%d",&n);
    for(int i=1;i<=n;++i)scanf("%d",&a[i]);
    scanf("%d%d",&x,&m);
    return;
}
int lower(C a,C b,int h)
{
    if(a.h>b.h)swap(a,b);
    if(!a.id)return b.id;
    if(!b.id)return a.id;
    return abs(a.h-h)<=abs(b.h-h)?a.id:b.id;
}
void init()
{
    int list1[MAXN],list2[MAXN],found[MAXN];
    C b[MAXN];
    for(int i=1;i<=n;++i)b[i].h=a[i],b[i].id=i;
    sort(b+1,b+1+n);
    for(int i=1;i<=n;++i)found[b[i].id]=i,list1[i]=i-1,list2[i]=i+1;
    list2[n]=0;
    for(int i=1;i<=n;++i)
    {
        int a=found[i],to=lower(b[list1[a]],b[list2[a]],b[a].h);
        if(to)addt(to,i,fa1,son1,bro1);
        int to2=to==b[list1[a]].id?list2[a]:list1[a];
        to2=lower(b[to2],b[found[lower(b[list1[list1[a]]],b[list2[list2[a]]],b[a].h))],b
[a].h);
        if(to2)addt(to2,i,fa2,son2,bro2);
        list2[list1[a]]=list1[a]?list2[a]:0;
        list1[list2[a]]=list2[a]?list1[a]:0;
    }
    for(int i=1;i<=n;++i)
    {
        f[i][0]=fa1[fa2[i]],
        dA[i][0]=abs(a[i]-a[fa2[i]]),dB[i][0]=abs(a[fa2[i]]-a[fa1[fa2[i]]]),
        d[i][0]=abs(a[i]-a[fa2[i]])+abs(a[fa2[i]]-a[fa1[fa2[i]]]);
    }
    for(int j=1;j<=16;++j)
        for(int i=1;i<=n;++i)
            f[i][j]=f[f[i][j-1]][j-1],
            dA[i][j]=dA[i][j-1]+dA[f[i][j-1]][j-1],dB[i][j]=dB[i][j-1]+dB[f[i][j-1]]
[j-1],d[i][j]=d[i][j-1]+d[f[i][j-1]][j-1];
    // for(int i=1;i<=n;++i)printf("%d ",fa2[i]);puts("");
    // for(int i=1;i<=n;++i)printf("%d ",abs(a[i]-a[fa2[i]]));puts("");
    // for(int i=1;i<=n;++i)printf("%d ",fa1[i]);puts("");
    // for(int i=1;i<=n;++i)printf("%d ",abs(a[i]-a[fa1[i]]));puts("");
}
void solve(int s,int x,int &jumpa,int &jumpb)
{
    jumpa=0,jumpb=0;
    for(int i=16;i>=0;--i)
    {
        if(!f[s][i]||d[s][i]>x)continue;
        // printf("%d-->%d,A:%d B:%d all:%d\n",s,f[s][i],dA[s][i],dB[s][i],d[s][i]);
        jumpa+=dA[s][i];
    }
}

```

```
        jumpb+=dB[s][i];
        x-=d[s][i];
        s=f[s][i];
    }
    if(fa2[s]&&abs(a[s]-a[fa2[s]])<=x) jumpa+=abs(a[s]-a[fa2[s]]);
}
int main()
{
    // freopen(".in","r",stdin);
    // freopen(".out","w",stdout);
    read();
    init();
    int ans=0;
    double v=1e9;
    for(int i=1;i<=n;++i)
    {
        int ls1,ls2;
        solve(i,x,ls1,ls2);
        if(ls2==0) {if(v==1e9&&a[ans]<a[i])ans=i;}
        else if((v==(double)ls1/ls2&&a[ans]<a[i])||(v>(double)ls1/
ls2))ans=i,v=(double)ls1/ls2;
    }
    printf("%d\n",ans);
    // printf("%d %lf\n",ans,v);
    while(m--)
    {
        scanf("%d%d",&s,&x);
        int ls1,ls2;
        solve(s,x,ls1,ls2);
        printf("%d %d\n",ls1,ls2);
    }
    return 0;
}
```

[<题目跳转>](#) [<查看代码>](#)