

const int N=1e5+100;

const int mod=9901;

using namespace std;

struct node

{

int v,id;

}a[N];

bool cmp(node a,node b)

{

return a.v<b.v;

}

int n,d,c[N],b[N];

int lowbit(int x)

{

return x&(-x);

}

void update(int x,int d)

{

while(x<=n)

{

c[x]+=d;

if(c[x]>mod)c[x]%=mod;

x+=lowbit(x);

}

}

int getsum(int x)

{

int ans=0;

while(x>0)

{

ans+=c[x];

if(ans>mod)ans%=mod;

x-=lowbit(x);

}

return ans;

}

int find(int x)

{

if(x>=a[n].v)return n;

if(x<a[1].v)return 0;

int l=1,r=n,ret=0;

while(l<=r)

{

int mid=(l+r)/2;

if(x>=a[mid].v)

{

ret=mid;

l=mid+1;

}

else

r=mid-1;

}

return ret;

}

int main()

{

while(scanf("%d%d",&n,&d)!=EOF)

{

memset(c,0,sizeof(c));

memset(b,0,sizeof(b));

for(int i=1;i<=n;i++)

{

scanf("%d",&a[i].v);

a[i].id=i;

}

sort(a+1,a+1+n,cmp);

for(int i=1;i<=n;i++)b[a[i].id]=i;

int sum=0;

for(int i=1;i<=n;i++)

{

int p=find(a[b[i]].v+d);

int q=find(a[b[i]].v-d-1);

int temp=getsum(p)-getsum(q);

temp=(temp+mod)%mod;

sum=(sum+temp)%mod;

update(b[i],temp+1);

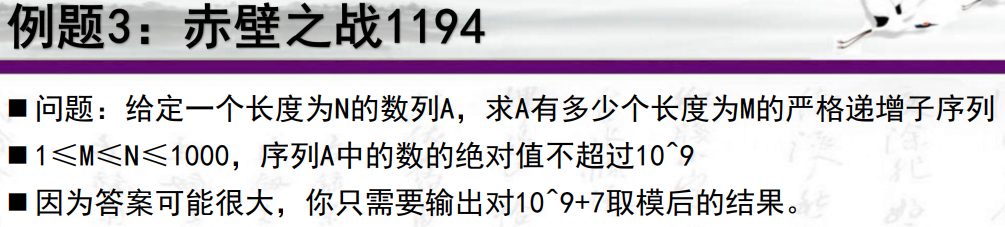
}

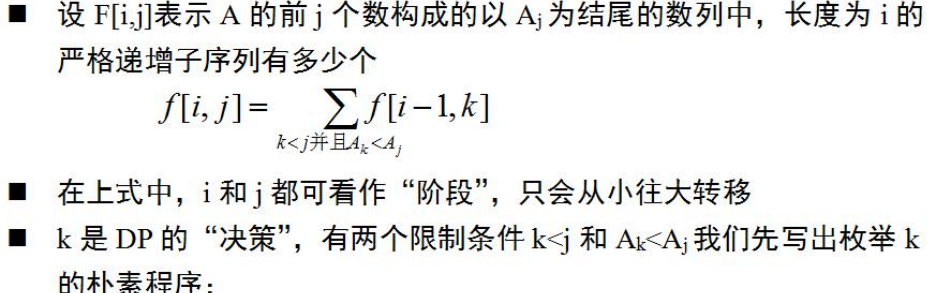
printf("%d\n",sum);

}

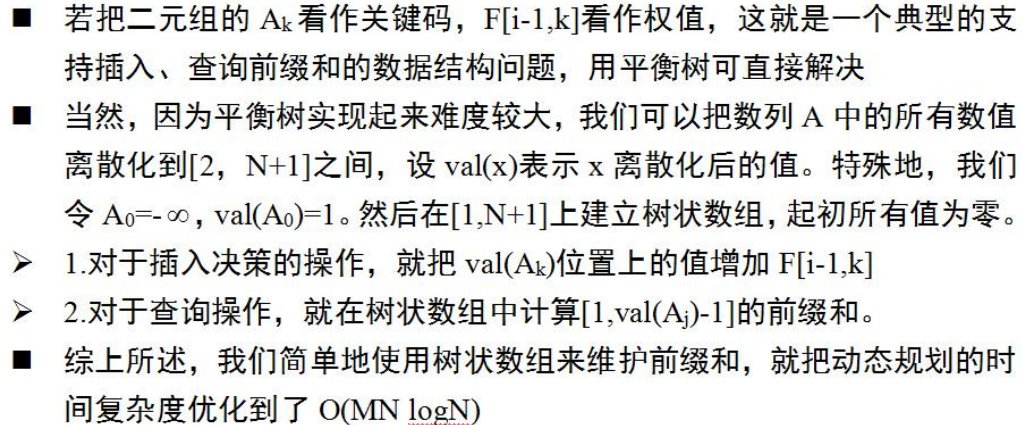
return 0;

}









#include<bits/stdc++.h>

using namespace std;

const int mod = 1e9 + 7;

struct node {

int a, id;

} e[1010];

int n, m;

int pos[1010];

int tree[1010];

int dp[1010][1010];

int ans, cnt;

inline int read() {

int x = 0, f = 1; char ch = getchar();

while (ch < '0' || ch > '9') {if (ch == '-') f = -1; ch = getchar();}

while (ch >= '0' && ch <= '9') {x = (x << 1) + (x << 3) + (ch ^ 48); ch = getchar();}

return x \* f;

}

int ask(int x) {

int sum = 0;

for (; x; x -= (x & -x)) sum += tree[x];

return sum;

}

void add(int x, int y) {

for (; x <= 1000; x += (x & -x)) tree[x] += y;

}

bool cmp(node a, node b) {

return a.a < b.a;

}

int main() {

int t = read();

while (t--) {

memset(dp, 0, sizeof dp);

ans = 0;

n = read() + 1, m = read();

for (int i = 2; i <= n; i++) e[i].a = read(), e[i].id = i;

sort(e + 2, e + n + 1, cmp);

for (int i = 2; i <= n; i++) pos[e[i].id] = i;

pos[1] = 1;

add(1, 1);

for (int k = 1; k <= m; k++) {

for (int i = 2; i <= n; i++) {

(dp[i][k] += ask(pos[i])) %= mod;

add(pos[i], dp[i][k - 1]);

}

memset(tree, 0, sizeof tree);

}

for (int i = 1; i <= n; i++) (ans += dp[i][m]) %= mod;

printf("Case #%d: %d\n", ++cnt, ans);

}

}