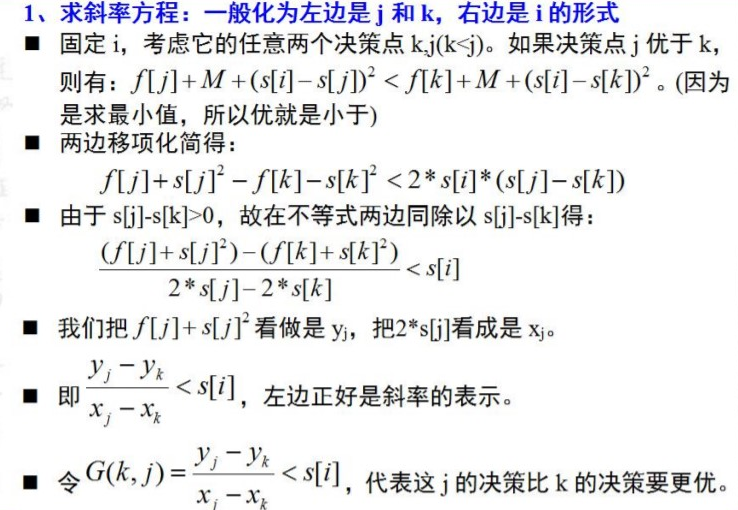
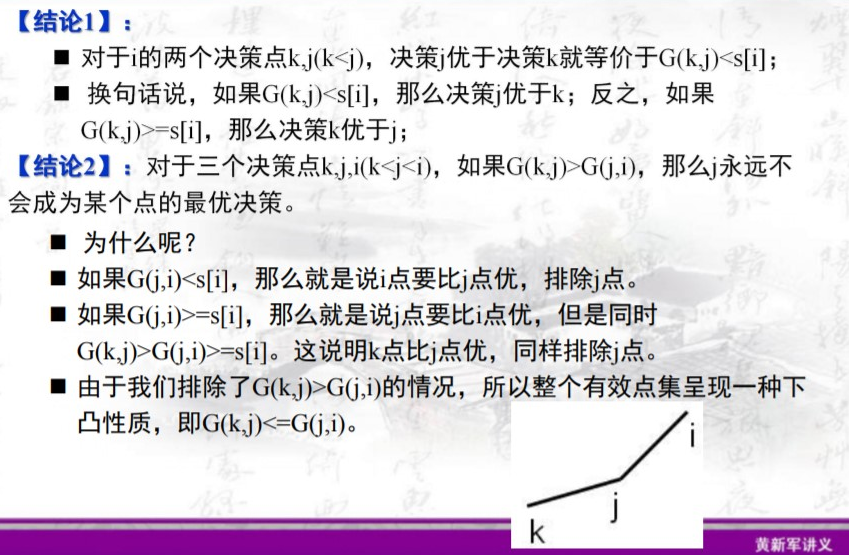
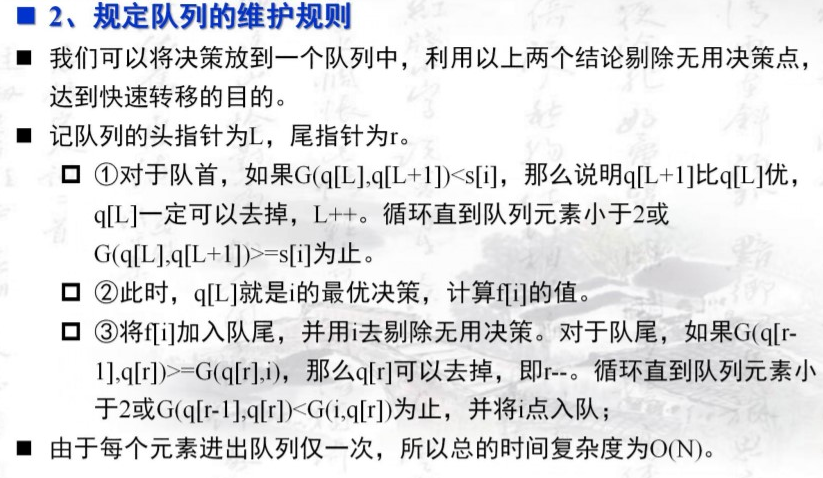


很显然，可以通过代数消元法来简化递推式从而得到类似于上面的这种典型式子







斜率dp的一般推法是，比如求f[i]的最小值，f[i]= F(j) 或者f[i]= F(k)，假设j比k优，

则F(j)< F(k)，推出一个不等式，然后带入到dp（）第一个while语句里面，使得q[l+1]比q[l]更优，这样才能l++，下面那个while语句取相反符号即可

#include<bits/stdc++.h>

#define ll long long

#define inf 1e18

#define mn 500005

using namespace std;

ll q[mn],s[mn]={0},f[mn],cnt=0,n,m;

double cal(ll i,ll j)

{double y=f[i]+s[i]\*s[i],y2=f[j]+s[j]\*s[j],x=2\*s[i],x2=2\*s[j];

return (y-y2)/(x-x2);

}

void dp()//一般情况下，两个while后面的比较符号是相反的

{ll i,j,l=1,r=1;

q[1]=0;f[0]=0;

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

{

while(l<r&&cal(q[l],q[l+1])<s[i])l++;

j=q[l];

f[i]=f[j]+(s[i]-s[j])\*(s[i]-s[j])+m;

while(l<r&&cal(q[r-1],q[r])>cal(q[r],i))r--;

q[++r]=i;

}

cout<<f[n]<<endl;

}

int main()

{

ll i,j,k,x,y,z;

char ch;

while(cin>>n>>m)

{

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

{

cin>>x;

s[i]=s[i-1]+x;

}

dp();

}

return 0;

}