1. 【题意】
2. 有n（500）本书，成一摞。
3. 我们有一个看书的顺序。
4. 要看第i本书的时候，
5. 我们要——
6. 1，把这本书之上的书搬走，体力成本加上这本书之上的书的重量之和，再放回之上的所有书（不改变顺序）；
7. 2，看这本书，把这本书放在最上面。
9. 问你初始的书从上到下的顺序可以是怎么样，能够使得我们的体力成本最小
11. 【类型】
12. 脑洞
14. 【分析】
15. 显然，我们书的初始摆放顺序就按照我们看书的顺序就好了。
16. 否则，我们还要浪费多余成本的调整成这个样子。
17. 于是按照这个原则贪心下去就可以得到最优解。
19. 【时间复杂度&&优化】
20. O(nm)

题意：一个人要读n本书每本书有个重量wi，要读m天，每天读1本，他每天把要看的书抽出来（把上面的搬开，拿出要读的书再把上面的书放回去），看完以后放到一摞书的最上面，问根据他的阅读顺序怎样初始化书的排列顺序能使他搬书的重量最小，求出这个最小重量。

思路：每本书第一次出现的次序即为书的排放次序，找到两次读同一本书之间的读过的书即可（减去重复的）

#include<cstdio>

#include<cstring>

#include<cmath>

#include<set>

#include<iostream>

#include<algorithm>

using namespace std;

bool vis[510];

int w[510];

int b[1010];

int main()

{

int n,m;

cin>>n>>m;

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

cin>>w[i];

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

cin>>b[i];

int ans(0);

for(int i=2;i<=m;i++)

{

memset(vis,0,sizeof(vis));

for(int j=i-1;j>0;j--)

{

if(b[i]==b[j])break;

if(vis[b[j]]==0)

{

ans+=w[b[j]];

vis[b[j]]=1;

}

}

}

cout<<ans<<"\n";

return 0;

}