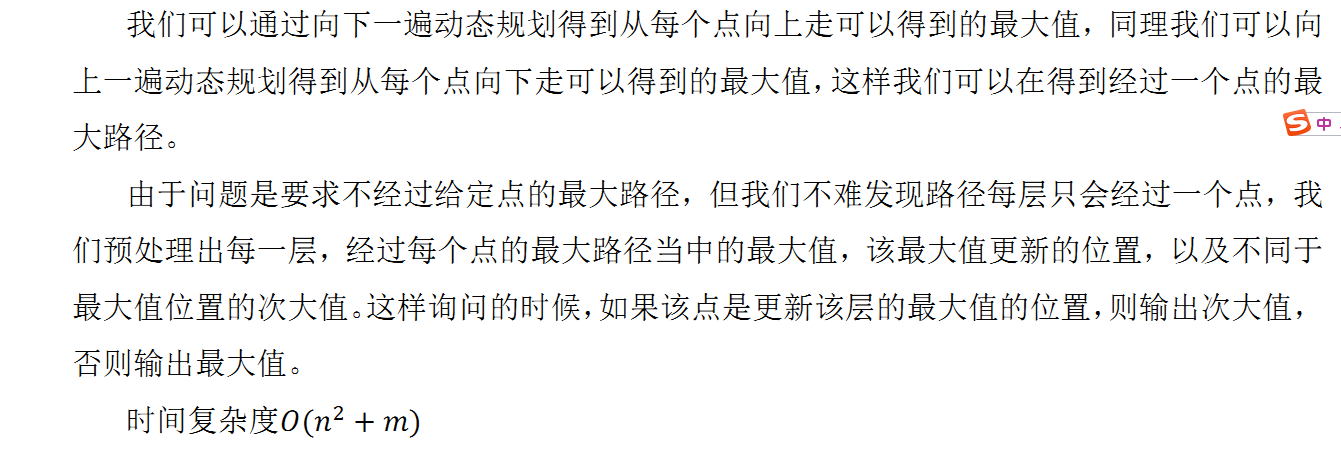
1



2

注意到m<=100，每个牛棚是独立于其他存在的。

假如某个牛棚有N头牛，该牛棚可以消除K次：

将代价序列分成(K+1)份，使每份的数目尽量接近。

那么一部分代价为1..[n/(K+1)]，另一部分为1..[n/(k+1)]+1

计算是显然的。

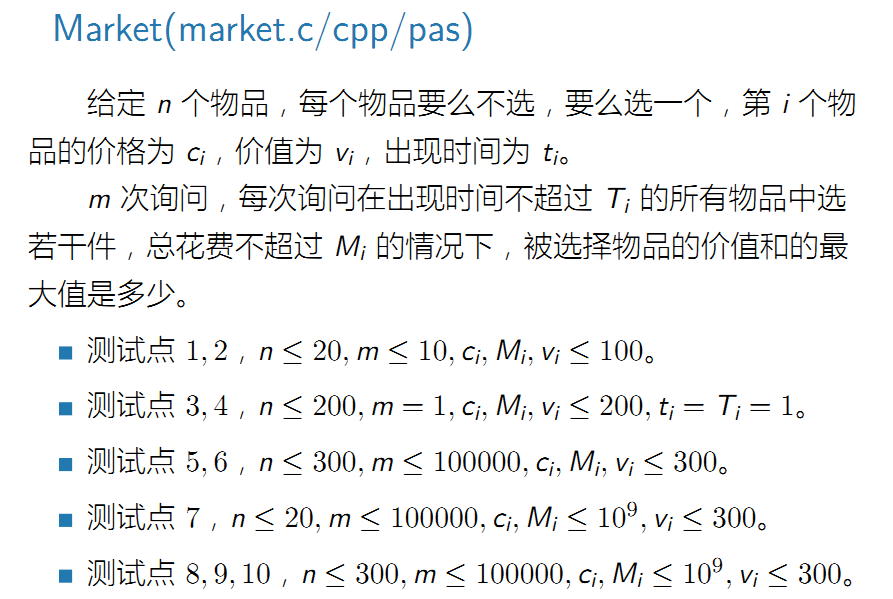
这一部分可以在O(1)的时间内算出。

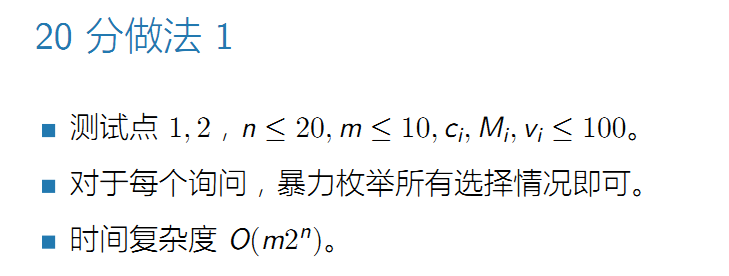
现在的问题转化为，M个牛棚，每个牛棚有P[I]头牛要进入，总共消除k次，使得最终答案最小。

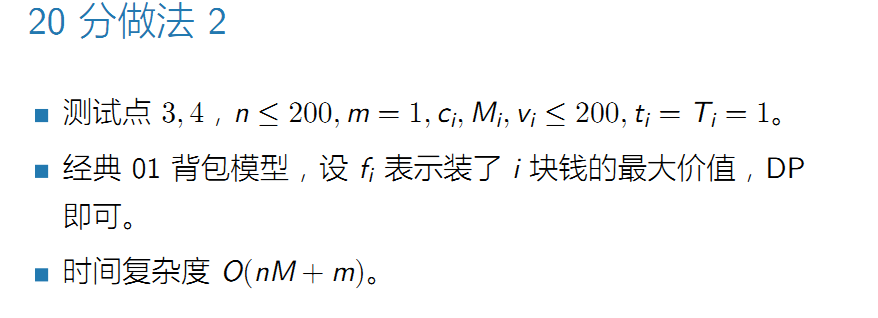
有:DP[i][j]=dp[i-1][j-k]+C(P[i],K)

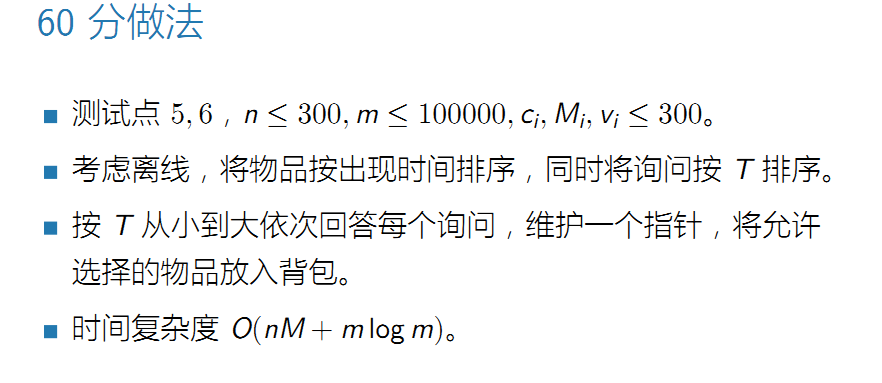
DP复杂度O(mk^2）。期望得分100.

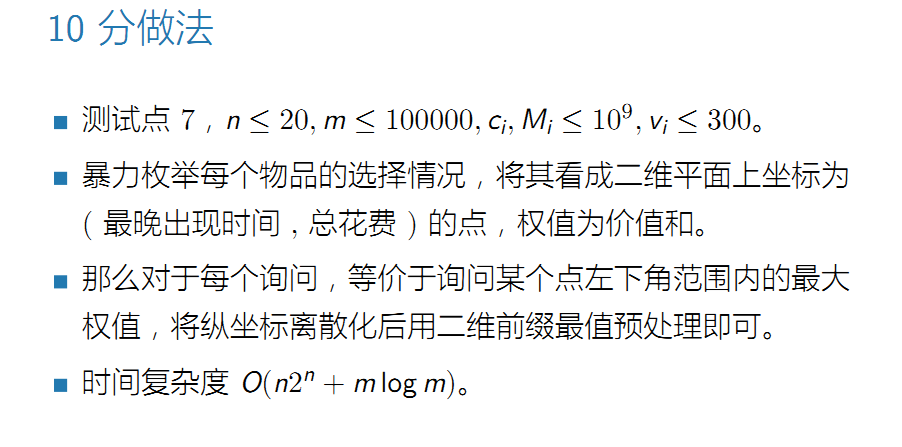
3、

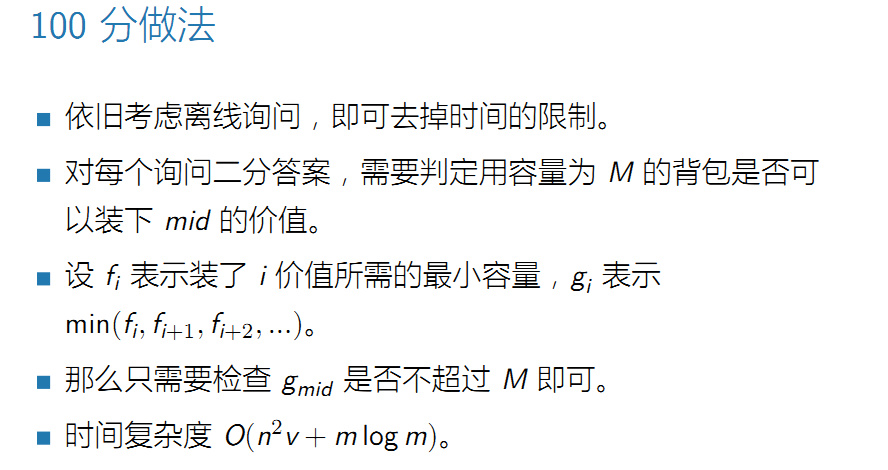




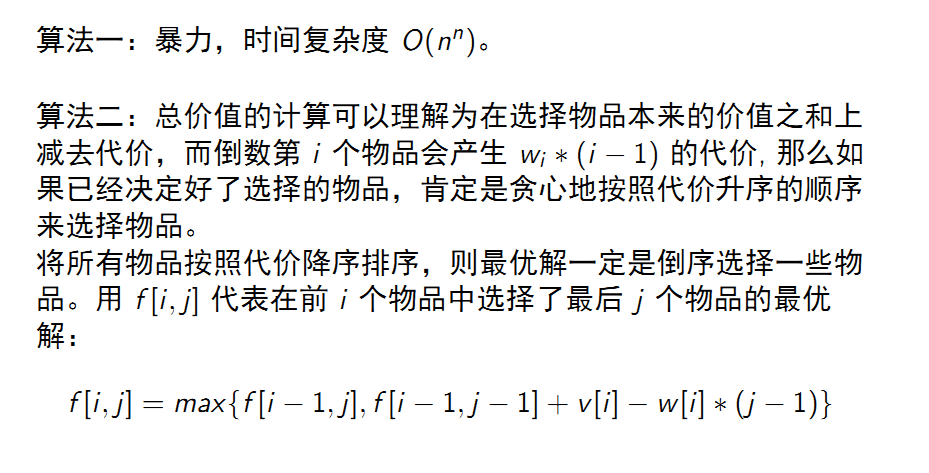








4、



5 usaco2010 gather

solution：

1、将树看成以1为根的一棵有根树。

2、用一个s[N]记录该节点以下的牛的个数。

3、用一个a[N]表示该节点为聚会地点时，奶牛们的不方便程度。

那么有：

s[ i ]=v[i]+s[ s(i) ];

a[ i ]+=a[ son(i) ]+s[son(i)]\*w

a[ i ]+=(total - s[ i ])\*w+(a[father(i) ]-a[ i ]-s[v]\*w)

其中v[i] 表示该点的牛数目，total 表示总的牛数量，w 表示 father(i) 到i点的距离。

这样，进行两次DFS，第一次DFS则要得到s[]的值还有以这棵树来看时候的a[]值，

这个时候a[]值仅为a[ son(i) ] + s[ son(i) ]\* w,

不算包括父亲在内的另一部分子树的不满意度，第二次DFS则要计算a[]，

需要两次DFS的原因是，必须要用一个节点的不满意度已经求出，才能求出这个点以下的节点的不满意度。

6、

