1、存放于磁带上文件需要顺序访问。故假设磁带上依次存储了*n*个长度分别是*L*[1],….,*L*[*n*]的文件，则访问第*k*个文件的代价为。

　　现给定*n*个文件的长度*L*[1],….,*L*[*n*]，并假设每个文件被访问的概率相等，试给出这*n*个文件在磁带上的存储顺序使得平均访问代价最小。

（1）简述算法采用的贪心策略。

（2）表述并证明问题的贪心选择性。

（3）表述并证明问题的优化子结构。

（4）写出算法的伪代码。

（5）分析算法的时间复杂度。

2、程序员接到*n*项编程任务，第*i*项任务需要在时间点*Ei*之前完成，完成第*i*项任务需要的时间（预计）为*ti*。每个任务顺利完成之后，程序员将得到固定的报酬a；如果未能按时完成某项任务，则程序员不能得到该任务的酬金。试设计一个贪心算法，为程序员安排工作计划，使其获得酬金最大。

（1）简述算法采用的贪心策略。

（2）表述并证明问题的贪心选择性。

（3）表述并证明问题的优化子结构。

（4）写出算法的伪代码。

（5）分析算法的时间复杂度。

（1）假设Q是所有文件的集合；

①找出Q中最短的文件k，在磁带上存储；

②随后Q=Q-{k}，返回步骤①，直到Q中没有文件为止。

（2）

贪心选择性：

设Q={1，2...n}是n个文件的集合，其中最短的文件为k(1<=k<=n)，把k写入磁带，则Q的文件选择问题的某个最优解A中，写入磁带的第一个文件一定是k；

证明：

假设Q的一个最优解为A，A即表示依次存储了*n*个长度分别是M[1],….,M[*n*]的文件；

若A中第一个存放的文件是k，证毕；

若A中第一个存放的文件不是k，设A中放在第一个的文件为m，此时，平均访问代价为



CA=



=

因此，A的最优解一定包含k；

（3）

优化子结构：

设Q={1，2...n}是n个文件的集合，文件的长度为*L*[1],….,*L*[*n*]；其中最短的文件为k（1<=k<=n)，设A是Q的一个最优解，A’=A-{k}是Q-{k}的一个最优解；

证明：

反证法：假如在集合Q-{k}上存在一个解B’（B’代表磁带上依次存储了*n*个长度分别是Z[1],….,Z[*n*]的文件）使得CB’<CA’，

则CA=CA’+W+C{k}；CB=CB’+W+C{k},其中W=(n-1)\*L[k]，C{k}=L[k];

此时显然有CB<CA，则我们的假设“A是Q的一个最优解”不成立，

因此在集合Q-{k}上不存在一个解B’使得使得CB’<CA’；

因此A’=A-{k}是Q-{k}的一个最优解，证毕；

（4）伪代码

假设数组L为一个有序字典，字典的key为各个事件，values即为文件长度；

BestOder(L):

对L按照values的值进行増序排序;

i=0

for key in L.keys() Do:

Order[i++]=key

return order

1. 时间复杂度：O(nlogn)

假设Q是所有任务的集合；Ei是最晚结束时间列表，D是一个用于暂时存放<事件n,时间T>键值对的双头列表，目前为空，res为存放结果的列表；

①对Ei进行降序排序。

②假设Ei列表中第一个为Ek，第二个为El。Ek对应的第k个事件的时间为tk，把数据对<事件k,tk>插入有序字典D，同时删除Ei列表中的Ek元素；另外，设Period=Ek-Ei。

③

若：对于有序字典D中T最短的事件m有 Tm<Period（即我们认为m事件可以在Period时间段内完成），则把事件m在D中删除，同时把事件m添加到结果列表res中，Period-=Tm；

否则：把<事件m,Tm-Period>（m为D中时间最短的事件）插入有序字典D末端，同时把D的<m,Tm>键值对删除，flag=0；

④ 若满足D不为空且flag!=0的条件，返回步骤③，否则运行步骤⑤

⑤若Ei不为空，返回第②步继续执行；否则运行⑥

⑥返回结果列表res

1. 贪心选择性：

我们假设贪心选择得到的为事件m（m满足:对于D列表中T最短的事件m有 Tm<Period，Period=El-Ek），m在El~Ek（当前Ei列表中最小的为Ek，次最小的为El）时间段内完成，则在Q的某个最优解A中，第一个完成的任务一定是m；

证明：

若A中第一个完成的任务是m，则证毕；

若A中第一个完成的任务不是m，则我们设A中第一个完成的事件为n（n∈D），我们假设欲完成某事件的最晚开始时间为Si=Ei-Ti；

B={m}∪A-{n}，则∀i∈A-{n}，Tn≥Tm，且Tn<Period（因为A也构成一个最优解）；则B也是Q的一个解，同时，|B|=|A|，因此B也是Q的一个最优解。

1. 优化子结构：

Q是n个事件的集合，假设Q的一个最优解为A，对Q中任务结束的时间Ei进行降序排序，假设结束最早的任务为k，Q-{k}的最优解 为A’=A-{k};

证明：

假设Q-{k}存在一个最优解B’，则有|B’|<|A’|，令B={k}∪B’,对于∀i∈{j∈Q|end<Sj}，

Si≥Sk，因此B中活动相容，B是Q的一个解；

由于|A|=|A’|+1, |B|=|B’|+1>|A’|+1=|A|,这与A是最优解的矛盾.

（4）伪代码：

假设E为一个有序字典，字典的key为各个任务，values即为任务结束时间；

t也是一个有序字典，字典的key为各个任务，values即为任务持续时间；

D是一个有序字典，字典的key为各个任务，values即为任务持续时间；

BestArrangement(E,t):

对E按照values的值进行降序排序;

While !E.isEmpty() :

Ei中第一个事件为k,第二个为l;

D[k]=t[k] #添加键值对

Period=E[k]-E[l]

E.pop(k) #弹出键值对

flag=1

while flag!=0 AND !D.isEmpty() :

D的数据对中，具有最小t的事件为m;

If D[m]<Period

THEN:

D.pop[m]

Period-=D[m]

res.append(m) #把m添加到返回结果中

ELSE:

temp=D.pop[m]

D[m]=temp-Period

flag=0

return res

（5）时间复杂度O(nlogn)