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。

将文件按照从小到大的顺序存储在磁带上。

（2）表述并证明问题的贪心选择性。

给定*n*个文件的长度分别是*L*[1],….,*L*[*n*]，其中访问第*k*个文件的代价为，每个文件被访问的概率相等，令，则存在一个存储方案，的平均访问代价是所有存储方案中最小的，而且中排在第一位.

证明：令是一个平均访问代价最小的存储方案，若在中排在第一位，则得证。

否则，令文件（长度最小的文件）在中排在第位，中文件长度依次为, , , ，其中第个文件的访问代价为。交换中第个文件和第个文件后，得到存储方案，往证的平均访问代价不大于。

的平均访问代价如下：

.

的平均访问代价如下：

.

由此得证具有最小的平均访问代价，而且中排在第一位.

（3）表述并证明问题的优化子结构。

给定包含*n*个文件的集合，其中文件的长度分别是*L*[1],….,*L*[*n*]，访问第*k*个文件的代价为，每个文件被访问的概率相等，令，令是平均访问代价最小的存储方案，而且中排在第一位，记，则是的平均访问代价最小的存储方案.

证明：记为存储代价的平均访问代价，则有

*.*

若不是的平均访问代价最小的存储方案，则存在是的平均访问代价最小的存储方案，通过与上式类似的推导，可以得到文件和组成的存储方案的平均访问代价为

。

与T是平均访问代价最小的存储方案矛盾，因此是的平均访问代价最小的存储方案。

（4）写出算法的伪代码。

略。

（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。

每次选择当前可以选择的(即)长度最小的任务，放入任务队列中，已经选择的任务按照最晚结束时间从小到大排序.

（2）表述并证明问题的贪心选择性。

设, , . 必然存在的最优调度, 包含, 并且中的任务按照最晚完成时间升序排列.

证明: 首先证明存在优化解包含任务. 令是的一个优化调度, 如果包含任务, 则得证优化解包含任务. 否则, 用任务替换中的第一个任务, 因为任务的长度不超过中第一个任务的长度,中的其余任务的完成时间不会后延, 于是得到一个新的调度, 其中包含任务数量和最优解相同, 因此也是一个最优调度, 并且包含任务.

若中的任务按照最晚完成时间从小到大排序, 则贪心选择性得证. 否则, 考虑P中任意相邻的两个任务和, 其中在前, 而且任务的完成时间必然不超过. 如果, 则交换任务和后, 其他任务的完成时间不变, 的完成时间提前, 的完成时间后延至交换前的完成时间. 由于交换前的完成时间必然不超过, 交换后必然能够按时完成, 其他任务显然也可以按时完成. 因此, 对于任意相邻的两个任务和, 其中在前, , 则交换任务和后, 所有任务依旧可以按时完成, 可调度的任务数量不变. 可以从开始, 不断地交换最晚完成时间逆序的相邻任务, 直至得到按照完成时间排序的调度, 此时中按时完成的任务数量与相同,也是优化解, 即存在优化解包含任务, 而且中的任务按照最晚完成时间排序.

（3）表述并证明问题的优化子结构。

设, , . 是的最优调度, 包含, 并且中的任务按照最晚完成时间升序排列.复制中的元素到, 修改中的元素, 对于: 如果, 则; 如果, 则; 如果, 则从中去掉, 即. 从中去掉, 其余任务顺序不变得到, 则是的最优调度, 而且中任务按照最晚完成时间升序排列.

证明: 易知中任务按照最晚完成时间升序排列. 如果不是的最优调度, 令是的最优调度, 而且中任务按照最晚完成时间升序排列. 于是. 将中最晚完成时间大于任务向后延时, 将得到三个部分： 中最晚完成时间不大于任务序列；长度为的空闲时间； 中最晚完成时间大于任务序列. 将置于空闲时间将得到一个的调度, 而且 , 与是的最优调度矛盾！因此是的最优调度, 而且中任务按照最晚完成时间升序排列.

（4）写出算法的伪代码。

Optimal-plan(A)

Input: A={(, ) | }

Output:

1.

2.

3. While

4.

5.

5. For

6. If

7.

8. Else if

9.

10. If

11.

12. 并将中元素按照升序排序

15. Return

（5）分析算法的时间复杂度。

.