算法实现题 4-2

给定 k 个排好序的序列 $s_1, s_2, \cdots\cdots, s_k$,用 2 路合并算法将这 k 个序列合并成一个序列。假设所采用的 2 路合并算法合并 2 个长度分别为 m 和 n 的序列需要 m+n-1 次比较。试设计一个算法确定合并这个序列的最优合并顺序，使所需的总比较次数最少。

为了进行比较，还需要确定合并这个序列的最差合并顺序，使所需的总比较次数最多。

本题是哈夫曼算法的应用，为了使总的比较次数最小，需要先对所有待合并序列的长度进行非降序排序，然后让序列长度最短的两个序列先进行合并，序列长度越长越靠后，如此构造哈夫曼树，序列长度最短的两个序列长度作为最深叶子节点，序列长度越长的越靠近根节点。

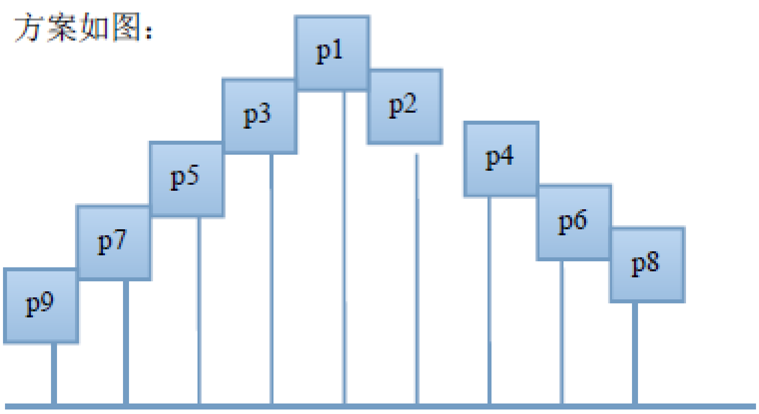
算法实现题 4-4

设磁盘上有 n 个文件 $f_1,f_2,\cdots,f_n$，每个文件占用磁盘上的 1 个磁道。这 n 个文件的检索概率分别是 $p_1,p_2,\cdots,p_n$，且$\sum_{i=1}^{n} p_i = 1$。磁头从当前磁道移到被检信息磁道所需的时间可用这 2 个磁道之间的径向距离来度量。如果文件 f_i 存放在第 i 道上，1≦i≦n，则检索这 n 个文件的期望时间是对于所有的 i<j,time+=p_i*p_j*d(i,j)。其中 d(i,j)是第 i 道与第 j 道之间的径向距离|i-j|。磁盘文件的最优存储问题要求确定这 n 个文件在磁盘上的存储位置，使期望检索时间达到最小。设计一个解此问题的算法，并分析算法的正确性与计算复杂性。

1、贪心选择策略

首先考虑目标函数：D=Σpi*pj*d(i,j)，其中对指定的 i 和 j，pi,pj 是不变的，可变因子是 d(i,j)。如果将一切 d(i,j)视为一个元素恒定的集合，即它们与文件的排列顺序无关，这就相当于给定了 1/2*n*(n-1)个元素 pipj 的集合 A 和 1/2n*(n-1)个元素 d(i,j)的集合 B，现在问题转换成从 A、B 两个集合中各选一个元素配对求积，使积的总和最小。设 a1>a2，a1a2 属于 A，b1b2 属于 B，则有以下两种配对方法：（a1b1,a2b2），（a1b2,a2b1）又：a1b2+a2b1< a1b1+a2b2。这说明要使 d(i,j)尽量小，则 pipj 尽量大。

将 n 个文件按概率大小进行降序排序，记 p1>=p2>=…>=pn，将 f2,f3 分别靠在 f1 左右两侧，接着 f4 在 f2 的右侧，以此类推，这种排列将是最佳方案。



2、贪心选择证明

此问题的一个最佳解必须满足以下条件：以任何两个磁道之间的中心线为界，落在中心线一边的各磁道的文件被检索的概率分别大于或小于另一边的概率。设有一条中心线 x，左侧是 C、A，右侧是 B、D。概率分别是 p2、p1、p3、p4。p1>p3,p2<p4。设 X=（X[K1],X[K2]…X[Kn]）是一个最佳解，且包含上述假设。现尝试第二种存储方案。除了 fc 和 fd 交换磁道外，其他均相同。

这时 Dk’(A,B,C,D)=p1p2*2d+p1p3*d+p1p4*d+p1p1*2d+p3p4*2d+p2p4*3d

$Dk=p1p2*d+p1p3*d+p1p4*2d+p2p3*2e+p3p4*d+p2p4*3d$

$Dk-Dk'=p1p2*d+p1p4*d+p3p4*d-p1p3*d=d(p4-p2)(p1-p3)>0$ ①

接下来，令

$H(k)=\Sigma pip2*d(i,fc)+\ \Sigma pip4*d(i,fd)+\ \Sigma pjp2*d(j,fc)+\ \Sigma pjp4*d(j,fd)$

$H(k')=\Sigma pip2*d(i,fc')+\ \Sigma pip4*d(i,fd')+\Sigma pjp2*d(j,fc')+\Sigma pjp4*d(j,fd')$

$H(K)-H(K')=-3p2\Sigma pi+3p4\Sigma pi+3p2\Sigma pj-3p4\Sigma pj=3(p4-p2)(\Sigma pi-\Sigma pj)>=0$ ②

由①和②可知，第二种方案并非最优解。所以方案一位最优解。

因此，磁道存储问题满足贪心选择性质。

3、最优子结构性质证明

按上述方法，设将 f1 放在 0 磁道上，则 f2 必须放在+1 或者-1 的位置上。若将 f2 放在 2 磁道上：

（1）如果Σpi>=Σpj，则交换 1、2 道的文件将获得更好的解。

（2）如果Σpi<=Σpj，显然交换 0、1 两道的文件会得到更好的解。

因此，当 f1 放在 0 道时，f2 必须放在+1 或-1 的位置上。

当 f1 放在 0 道，f2 放在 1 道，可以证明 f3 不放在-1 道。若 f3 放在-2 或+2，都不是最佳的。

综上，可以归纳证明，此问题满足最优子结构性质。

4、算法复杂度分析

①对检索概率进行排序采用快排的方式，那么算法复杂度是 O(n log n)

②对各个文件根据概率大小从中间向外面摆放，这个操作的算法复杂度是 O(n)

所以，该算法的时间复杂度为 O(n log n)。

算法实现题 4-6

设有 n 个顾客同时等待一项服务。顾客 i 需要的服务时间为 ti, 1≦i ≦n 。共有 s 处可以提供此服务。应如何安排 n 个顾客的服务次序才能使平均等待时间达到最小？平均等待时间是 n 个顾客等待服务时间的总和除以 n。

1、贪心策略

先对所有顾客的所需要的服务时间 ti 进行非降序排序，然后让服务时间最短的顾客先接受服务，服务时间越长越靠后。

2、贪心选择性质证明（略）

3、最优子结构性质证明（略）

4、算法复杂度分析

对所有顾客的所需要的服务时间 ti 进行非降序排序的时间复杂度为 O(n log n)，之后计算平均服务时间需要遍历一遍所有顾客，时间为 O(n)，所以算法的整体复杂性为 O(nlogn)。

算法实现题 4-9

一辆汽车加满油后可行驶 n 公里。旅途中有若干个加油站。设计一个有效算法，指出应在哪些加油站停靠加油，使沿途加油次数最少。对于给定的 n(n <= 5000)和 k(k <= 1000)个加油站位置，编程计算最少加油次数。并证明算法能产生一个最优解。

1、贪心策略

要使汽车加油次数最少，就必须让汽车跑的尽可能远，话贪心选择最远的加油站进行加油，用这种贪心的策略。

2、贪心选择性质证明

该题设在加满油后可行驶的 N 千米这段路程上任取两个加油站 A、B，且 A 距离始点比

B距离始点近，则若在B加油不能到达终点那么在A加油一定不能到达终点，因为$m+N<n+N$，即在B 点加油可行驶的路程比在A 点加油可行驶的路程要长n-m 千米，所以只要终点不在B、C之间且在C 的右边的话，根据贪心选择，为使加油次数最少就会选择距离加满油得点远一些的加油站去加油，因此，加油次数最少满足贪心选择性质。

3、最优子结构性质证明

在执行了上述的贪心选择策略的时候，我们已经在d(s)处进行了加油，那么现在问题就变成跟之前问题有同样性质的子问题：第m 个加油站到第k 个加油站之间汽车加油次数最少。设总体的加油次数为T，则T(1,k)=1+T(s,k)，现在我们用反证法来证明最优子结构性质：

假设T(1,k)是最优解，那么假设存在一个子问题的更优的加油次数T’，使得$T'<T(s,k)$，那么加上定值后呢会得到一个值T’(1,k)，使得$T'(1,k)<T(1,k)$，这与T(1,k)是最优值矛盾。所以该问题具有最优子结构性质。

4、算法复杂度分析

只需要遍历整个加油站之间距离的数组，所以复杂度是O(n)。

算法实现题4-11

给定n 位正整数a，去掉其中任意k≤n 个数字后，剩下的数字按原次序排列组成一个新的正整数。对于给定的n 位正整数a 和正整数k，设计一个算法找出剩下数字组成的新数最小的删数方案。

1、贪心策略

设本问题为T。最优解A=（y1,y2…yk）表示依次删去的k 个数。在删去k个数后剩下的数字按原次序排成的新数，最优值记为TA。

求解采用最近下降点有限的贪心策略：$x1<x2<\cdots xi<xj$；如果$xk<xj$，则删去xj，得到一个新的数，且n-1 位中最小的数N1 可表示为x1x2…xixk…xn。对N1 而言，删去了一位数后，原问题变成了需对n-1 位数删去k-1 个数的新问题T’。新问题和原问题相同，只是问题规模由n 减为n-1。基于这种删除策略，对新问题，选择最近下降点的数进行删除，直至删除k 个数。

2、贪心选择性质证明

根据数的进制特点，对a 按权展开的：

a=x1*10^n-1+x2*10^n-2+…xi*10^n-i+xj*10^n-j+xk*10^n-k+…xn

有：N1= x1*10^n-2+x2*10^n-3+…xi*10^n-i-1+xk*10^n-k+…xn

假设删去的不是xj 而是其他位，

则有：N2= x1*10^n-2+x2*10^n-3+…xi*10^n-i-1+xj*10^n-k+…xn

因为$x1<x2<\cdots xi<xj$，且xj>xk，所以$N1<N2$

所以满足贪心选择性质。

3、最优子结构性质证明

假设A’不是子问题的最优解，其子问题的最优解为B’，其最优值为TB’，则$TB'<TA'$，而TA=TA’+xj*10^n-j，且$TB'<TA'$，所以：

$TB'+xj*10^n-j<TA'+xj*10^n-j$。即存在一个由数a 删去1 位数后得到的n-1 位数比最优值TA 更小。这与TA 为问题T 的最优值相矛盾。所以A’是子问题T’的最优值。

因此满足最优子结构性质。

4、算法复杂度分析

若这个数有n 位，则每删除一位需要按高位到低位的顺序搜索一遍它的所有位，删除k位共需要遍历k 次，所以此算法的时间复杂度为O(k*n)。

算法实现题 4-15

设 n 是一个正整数。现在要求将 n 分解为若干互不相同的自然数的和，且使这些自然数的乘积最大。

1、贪心策略

将 n 分成从 2 开始的连续自然数的和。如果最后剩下一个数，将此数在后项优先的方式下均匀地分给前面各项。

2、贪心选择性质证明

先对整数分解分析可以发现如下结论：

若 a + b = const，则|a - b| 越小，a*b 越大。

对于 n < 4，可以验证其分解成几个正整数的和的乘积是小于 n 的。

对于 n >= 4, 能证明其能分解成几个数的和使得乘积不小于 n。

如果分解成 1 和 n - 1，那么对乘积是没有帮助的，因此，假设 n 分解成 a 和 n - a，2 <= a <= n - 2，那么

a * (n - a) - n
= (a - 1) * n - a * a + a - a
= (a - 1) * (n - a) - a
>= (a - 1) * 2 - a
= a - 2
>= 0

因为每次分解都能使乘积增加，而又要求分成互不相同的自然数，|a - b| 越小，a*b 越大，所以最优解必是最终分解结果。

3、最优子结构性质证明

设 x=2*3*…m*…*q 是最优分解，y=m*…*q，则 x=2*3*……*y，若 y 不是最优分解，则存在 y2>y，则存在 x2=2*3*……*y2>x，与假设 x 为最优解矛盾，故 y 也是最优分解。

4、算法复杂度分析

本题主要的复杂度就是一层循环内从 2 开始累加自然数，直到和大过要拆分的自然数 n，假设累加到的自然数是 t，则（2+t）*（t-1）/2 >=n,可推出 t>= n ，也就是说本题的复杂度是$O(\sqrt{n})$。