**算法设计与分析**

一、一百四十页

7.

a. 对于一个包含100万随机数的数组排序，快速排序比插入排序快多少倍?

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 常用排序法 | | | | |
| 排序法 | 最坏所需时间 | 平均所需时间 | 稳定性 | 所需的额外空间 |
| 直接插入 | O(n2) | O(n2) | 是 | 0(1) |
| 希尔排序 | O(n2) | O(n1.3) | 否 | 0(1) |
| 冒泡排序 | O(n2) | 9(n2) | 是 | 0(1) |
| 快速排序 | O(n2) | O(nlog2n) | 否 | O(nlog2n) |
| 直接选择排序 | O(n2)) | O(n2) | 是 | 0(1) |
| 堆排序 | (nlog2n) | O(nlog2n) | 否 | O(1) |
| 归并排序 | O(nlog2n) | O(nlog2n) | 是 | O(n) |

由常用排序法表中得知：

因为：

1. 插入排序的时间复杂度为O(n2)；
2. 快速排序的时间复杂度为O(nlog2n)
3. 故：插入排序的时间复杂度为/快速排序的时间复杂度为：

O(n2)/O(nlog2n)=n/log2n

当n=100 0000时：n/log2n=100 0000/log2100 0000

b.是非题:对于n>1的n元素数组，是否存在插入排序比快速排序更快的情形?

存在：当快速排序完成编译时间是属于最坏所需时间O(n2)，所需的额外空间是O(nlog2n)，差入排序的时间复杂度属于O(n2)，所需的额外空间为0(1)此时，当n>1的n元素数组存在插入排序比快速排序更快

二、225页 第六题

1. 切割木棍问题为下列问题设计一个动态规划算法。已知小木棍的销售价格p;和长度i相关，i=1,2,..n,如何把长度为n的木棍切割为若干根长度为整数的小木棍，使得所能获得的总销售价格最大?该算法的时间效率和空间效率各是多少？

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| i | 1 | 2 | 3 | 4 | 5 | ... | n |
| Pi | 5 | 7 | 3 | 8 | 9 | ... | Pn |

由题意得知木棍的长度为i（i=1，2，3，...,n）

由表格假设数据得知木棍长度i的价格Pi（Pi=5，7，3，8，9，...,Pn）

得长度为n的最大价值W(n)=MAX(price(i)+price(n-i))

长度为n的价格有两种:

第一种：原始长度为n时的价格W=Pn

第二种：加n分割为个小块再加起来的价格

W[i][P]=max{f[i-1][P],f[i-1][V]-P[i]]+W[i]}

下面的表格是假设有一个长度为五的棍子怎样切才能最大价值，得知为（1，4）是价值最大为13

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| i | Pi | Wi | N=1 | 2 | 3 | 4 | 5 |
| 1 | 5 | 1 | 5 | 5 | 5 | 5 | 5 |
| 2 | 7 | 2 | 5 | 7 | 12 | 12 | 12 |
| 3 | 3 | 3 | 5 | 7 | 12 | 12 | 12 |
| 4 | 8 | 4 | 5 | 7 | 12 | 12 | 13 |
| 5 | 9 | 5 | 5 | 7 | 12 | 12 | 13 |

先从最短的长度 1开始找相对应长度可得到的最大价值，因为长度1无法再分，所以W[1] 就为原始长度价格 p[1]

然后长度2的可得到的最大价值W[2]就为W[1] +W[1] 和 p[2]之中最大的那个。3的可得到的最大价值W[3]就为W[1]+W[2] 、W[2]+W[1]和p[3]中最大的那个。因为比当前长度小的所有整数长度的对应的最大价格都是已知的，所以长度为n时只需要找到W[1]+W[n-1]、W[2]+W[n-2]、…、W[i]+W[n-i]、…、W[n-1]+W[1]、p[n]中最大的值，再赋值给W[n]

此算法的时间效率是O(N^2)，空间效率是O (N)

三、229 第三题

3.

对于背包问题的自底向上动态规划算法，请证明:

a.它的时间效率属于O(nW)。

b.它的空间效率属于0(nW)。

c.从一张填好的动态规划表中求得最优子集的组合所用的时间属于O(n)。

解答：

背包问题:有n个物品，第i哥物品的体积为Vi，重量为Wi，对应价值为P1、、P2、P3、...Pn。

在前N件物品中，选取若干件物品放入背包中;

状态是:在前N件物品中，选取若干件物品放入所剩空间为W的背包中的所能获得的最大价值;

决策是:第N件物品放或者不放;

由此可以写出动态转移方程:

f[i,j]表示在前i件物品中选择若干件放在所剩空间为j的背包里所能获得的最大价值

F[i,j]=max{f[i-1,j]-Wi]+Pi (j>=Wi), f[i-1,j]}

这样，我们可以自底向\_上地得出在前M件物品中取出若干件放进背包能获得的最大价值，也就是f[n,w]

算法如下：

Begin

For i=0 to w do

F[0,j]=0;

For i=1 to n do

For j=0 to w do begin

F[i,j]=f[i-1,j]

If (j>=w) and if (j>=w) and (f[i-1,j-w]+Pi>f[i,j]) then

f[i,j]:=f[i-1,j-w]+Pi;

end

　ruturn(f[m,w]);

　　end;

得证：由于是用了一个二重循环，这个算法的时间复杂度是O（n\*w)

B、

背包问题，特点是每种物品仅有一件，可以选择放或不放。

　　f[v]表示前i件物品恰放入一个容量为v的背包可以获得的最大价值。则其[状态转移方程](https://baike.sogou.com/lemma/ShowInnerLink.htm?lemmaId=37255466&ss_c=ssc.citiao.link" \t "https://baike.sogou.com/_blank)便是：

　　f[v]=max{f[v],f[v-c]+w}

　　这个方程基本上所有跟背包相关的问题的方程都是由它衍生出来的。所以有必要将它详细解释一下：“将前i件物品放入容量为v的背包中”这个子问题，若只考虑第i件物品的策略（放或不放），那么就可以转化为一个只牵扯前i-1件物品的问题。如果不放第i件物品，那么问题就转化为“前i-1件物品放入容量为v的背包中”，价值为f[v]；如果放第i件物品，那么问题就转化为“前i-1件物品放入剩下的容量为v-c的背包中”，此时能获得的最大价值就是f[v-c]再加上通过放入第i件物品获得的价值w。

以上方法的[时间和空间](https://baike.sogou.com/lemma/ShowInnerLink.htm?lemmaId=9264164&ss_c=ssc.citiao.link" \t "https://baike.sogou.com/_blank)复杂度均为O(N\*w)

四、234页 第十一题

11. 矩阵连乘

考虑如何使得在计算n个矩阵的乘积A1 A2...An 时，总的乘法次数最小，这些矩阵的维度分别为d0\*d1,d1\*d2,..，dn-1\*dn,。假设所有两个矩阵的中间乘积都使用蛮力算法(基于定义)计算。

1. 给出一个三个矩阵连乘的例子，当分别用(A1A2)A3 和A1(A2A3)计算时，它们的乘法次数至少相差1 000倍。
2. 有多少种不同的方法来计算n个矩阵的连乘乘积?

c.设计一个求n个矩阵乘法最优次数的动态规划算法。

A:

矩阵乘法不满足分配律，但满足结合律，因此(A\* B)\*C既可以按顺序(A\*B)\*C进行，也可以按A\*(B\*C)来进行。

可得例子：A1：5\*1000、A2：1000\*5、A3：5\*1000

B:

算法数递推公式为，并且m(1)=1

C:

给出n个矩阵组成的序列，设计一种方法把它们依次相乘，使得总运算量最小。假设第i个矩阵Ai;是pi-1\*p;的

设f(i,j)表示把 Ai, Ai+1,...,A 乘起来所需要的乘法次数，枚举“最后一次乘法”是第k个乘号，则f(i,j)=max{f(i,k)+f(k+1,j)+pi-1pkpj},边界是f(i,i)=0,时间复杂度为O(n3)。

五、249 .7

7.谣言传播

有n个人，每个人都拥有不同的谣言。通过发电子信息，他们想互相

共享所有的谣言。假定发送者会在信息中包含他已知的所有谣言，而且一条信息只有一个收信人。设计一个贪心算法，保证在每个人都能获得所有谣言的条件下，使发送的信息数最小。

解答:

按照1发信给2, 2发信给3, 3发信给4，…，

n-1发信给n的方式发送谣言，该贪心算法基于每次发信都使得当前收信人掌握的谣言更多，最后由n将所有谣言发送给其他n-1个人。

发送信息总数为2n-2，这是最小的发信息数。因为每增加一个人，至少需要增加两次发送信息，当n=2是，发送信息数为2，归纳法可证明2n-2为最小发信息数

六、264 第九题

9.

a.写一个程序，为给定的英文文本构造一套哈夫曼编码，并对该文本编码。

b.写一个程序，对一段用哈夫曼码编码的英文文本进行解码

c.做一个实验，测试对包含1000个词的一段英文文本进行哈夫曼编码时，典型的压缩率位于什么样的区间。

d.对编码程序做一个实验，测试如果用标准的估计频率代替英文文本中字符的实际出现频率，该程序的压缩率会有什么样的变化。

七、331 第七题

9.a.写一个程序，为给定的英文文本构造一套哈夫曼编码，并对该文本编码。

b.写一个程序，对一段用哈夫曼码编码的英文文本进行解码。

c.做一个实验，测试对包含1000个词的一段英文文本进行哈夫曼编码时，典

型的压缩率位于什么样的区间。

d.对编码程序做一个实验，测试如果用标准的估计频率代替英文文本中字符的

实际出现频率，该程序的压缩率会有什么样的变化。

八、338 第七题

1. 用回溯法生成{1, 2,3,4}的所有排列。

#include<stdio.h>

#include<iostream>

using namespace std;

int a[4]={1,2,3,4};

void swap(int a[],int m,int n){

int k=a[m];

a[m]=a[n];

a[n]=k;

}

void qpl(int a[],int k,int m){

if(k==m){//如果待全排列数组，起始等于末尾，说明已经产生了一种结果，进行输出

for(int i=0;i<=k;i++){

cout<<a[i]<<" ";

}

cout<<endl;

}

else{

for(int i=k;i<=m;i++){

swap(a,i,k);//交换当前元素与i位置的元素

qpl(a,k+1,m);//对当前元素之后的元素进行全排列

swap(a,i,k);//回溯

}

}

}

int main(){

qpl(a,0,3); //diao yong fuction(qpl)

return 0;

}