第8周作业

总分：100分

得分：100.0分

有关随机化算法错误的是（）

A. 随机化算法的特征是对所求解问题的同一实例用同一随机化算法求解两次可能得到完全不同的效果，这两次求解问题所需的时间甚至所得到的结果可能会有相当大的差别。

B. 数值随机化算法常用于数值问题的求解，所得到的解都是精确解。

C. 蒙特卡罗算法用于求问题的准确解，但解不一定正确。

D. 舍伍德算法引入随机性来降低最坏情况出现的概率，从而消除或减少问题好坏实例之间的时间消耗的差异。

**回答正确**

**答案**

数值随机化算法常用于数值问题的求解，所得到的解都是精确解。

有关估算π值的随机化算法说法错误的是（）

A. 估算π值的随机化算法估算的近似值的精度随算法消耗的时间的增加而提高

B. 估算π值的随机化算法随机实验次数越多，估算的π值精度越高

C. 估算π值的随机化算法是数值随机化算法。

D. 估算π值的随机化算法估算的近似值的精度与算法消耗的时间无关

**回答正确**

**答案**

估算π值的随机化算法估算的近似值的精度与算法消耗的时间无关

有关主元素问题的蒙特卡罗算法说法错误的是（）

A. 主元素问题的蒙特卡罗算法每次执行都返回True 或False，True表示有主元素，False表示没有主元素。

B. 主元素问题的蒙特卡罗算法返回True的解是正确解，False的解不一定是正确解。

C. 主元素问题的蒙特卡罗算法得到正确解的概率随算法消耗的时间的增加而降低 。

D. 主元素问题的蒙特卡罗算法得到的解为正确解的概率大于0.5 。

**回答正确**

**答案**

主元素问题的蒙特卡罗算法得到正确解的概率随算法消耗的时间的增加而降低 。

有关素数测试问题算法说法正确的是（）

A. 根据Wilson定理，可以设计素数测试的随机化算法。

B. 可以采用试除法，设计素数测试的随机化算法。

C. 根据二次探测定理设计的素数测试蒙特卡罗算法得到的解为正确解的概率大于0.5

D. 根据二次探测定理，可以设计素数测试的蒙特卡罗算法，当算法返回True时，解一定正确；当返回False时，解不一定正确。

**回答正确**

**答案**

根据二次探测定理设计的素数测试蒙特卡罗算法得到的解为正确解的概率大于0.5

有关n皇后问题的拉斯维加斯算法说法正确的是（）

A. n皇后问题的拉斯维加斯算法可以采用对不冲突的多个列位置进行随机。

B. n皇后问题的拉斯维加斯算法得到接的概率小于0。

C. n皇后问题的拉斯维加斯算法每次运行都能得到一种n个皇后的放置方案。

D. 多次运行n皇后问题的拉斯维加斯算法并不能提高算法得到解的概率。

**回答正确**

**答案**

n皇后问题的拉斯维加斯算法可以采用对不冲突的多个列位置进行随机。

有关随机快速排序算法说法错误的是（）

A. 随机快速排序与快速排序的区别是随机快速排序随机选择基准元素，而快速排序的确定性算法选择固定位置的元素作为基准元素。

B. 随机快速排序通过对快速排序引入随机性，降低了快速排序最好和最坏情况出现的概率。

C. 随机快速排序的时间复杂度趋于O(nlogn)。

D. 随机快速排序每次运行都能够得到解，但是得到的解不一定正确。

**回答正确**

**答案**

随机快速排序每次运行都能够得到解，但是得到的解不一定正确。

有关整数n的因子分解问题说法正确的是（）

A. 整数的因子分解就是将整数n分解多个因子的乘积，并不要求因子的素数性。

B. 整数的因子分解问题不可以转化为因子分割问题。

C. 因子分割不可以采用试除法找出整数n的因子。

D. Pollard算法，只要给足够的时间，肯定能找到整数n的因子。

**回答正确**

**答案**

Pollard算法，只要给足够的时间，肯定能找到整数n的因子。

以下有关随机数产生的线性同余法说法正确的是（）

A. 线性同余法产生的随机数是伪随机数。

B. 线性同余法的系数是模数的倍数时，随机数的随机性能好。

C. 线性同余法的系数、增量、模数越大，随机数的随机性能越差。

D. 线性同余法的系数与模数互质，随机数的随机性能差。

**回答正确**

**答案**

线性同余法产生的随机数是伪随机数。

以下有关随机选择第k小算法正确的是（）

A. 随机选择第k小算法中的随机性和随机快速排序的随机性一样，都是随机选择基准元素。

B. 随机选择第k小算法是对线性时间选择算法中划分过程进行了随机，其他和线性时间选择算法一样。

C. 随机选择第k小算法划分过程结束后，要在比基准元素小的子问题中查找第k小。

D. 随机选择第k小算法中的随机性和随机快速排序的随机性不同，随机快速排序是随机选择基准元素，随机选择第k小算法随机划分、比较。

**回答正确**

**答案**

随机选择第k小算法中的随机性和随机快速排序的随机性一样，都是随机选择基准元素。

通过多次执行的方式提高随机算法得到正确解的概率的算法是（）

A. 数值随机化算法

B. 蒙特卡罗算法

C. 拉斯维加斯算法

D. 舍伍德算法

**回答正确**

**答案**

蒙特卡罗算法

以下算法中，通过多次执行能够提高算法得到解的概率的算法是（）

A. 拉斯维加斯算法

B. 舍伍德算法

C. 蒙特卡罗算法

D. 数值随机化算法

**回答正确**

**答案**

拉斯维加斯算法

以下算法中，哪个算法用于求问题的近似解，求得近似解的精确程度与算法消耗的时间相关（）

A. 蒙特卡罗算法

B. 拉斯维加斯算法

C. 数值随机化算法

D. 舍伍德算法

**回答正确**

**答案**

数值随机化算法

有关随机化算法正确的是（）

A. 随机化算法的特征是对所求解问题的同一实例用同一随机化算法求解两次可能得到完全不同的效果，这两次求解问题所需的时间甚至所得到的结果可能会有相当大的差别。

B. 数值随机化算法常用于数值问题的求解，所得到的解往往都是近似解，而且近似解的精度随计算时间的增加不断提高。

C. 蒙特卡罗算法用于求问题的准确解，但解不一定正确。

D. 拉斯维加斯算法绝不返回错误的解，但有时得不到问题的解。可以通过多次执行提高算法得到解的概率。

E. 舍伍德算法用于当一个确定性算法在最坏情况下的计算时间复杂性与其在平均情况下的计算复杂性有较大差异时。

F. 舍伍德算法引入随机性来降低最坏情况出现的概率，从而消除或减少问题好坏实例之间的时间消耗的差异。

**回答正确**

**答案**

随机化算法的特征是对所求解问题的同一实例用同一随机化算法求解两次可能得到完全不同的效果，这两次求解问题所需的时间甚至所得到的结果可能会有相当大的差别。

数值随机化算法常用于数值问题的求解，所得到的解往往都是近似解，而且近似解的精度随计算时间的增加不断提高。

蒙特卡罗算法用于求问题的准确解，但解不一定正确。

拉斯维加斯算法绝不返回错误的解，但有时得不到问题的解。可以通过多次执行提高算法得到解的概率。

舍伍德算法用于当一个确定性算法在最坏情况下的计算时间复杂性与其在平均情况下的计算复杂性有较大差异时。

舍伍德算法引入随机性来降低最坏情况出现的概率，从而消除或减少问题好坏实例之间的时间消耗的差异。

有关估算π值的随机化算法说法错误的是（）

A. 估算π值的随机化算法估算的精度随算法消耗的时间的增加而提高

B. 估算π值的随机化算法随机实验次数越多，估算的π值精度越高

C. 估算π值的随机化算法随机实验次数越多，估算的π值精度越低

D. 估算π值的随机化算法估算的近似值的精度随算法消耗的时间的增加而降低

**回答正确**

**答案**

估算π值的随机化算法随机实验次数越多，估算的π值精度越低

估算π值的随机化算法估算的近似值的精度随算法消耗的时间的增加而降低

有关随机快速排序算法说法正确的是（）

A. 随机快速排序与快速排序的区别是随机快速排序随机选择基准元素，而快速排序的确定性算法选择固定位置的元素作为基准元素。

B. 随机快速排序通过对快速排序引入随机性，降低了快速排序最好情况出现的概率。

C. 随机快速排序的时间复杂度趋于O(nlogn)。

D. 随机快速排序每次运行都能够得到解，但解不一定正确。

**回答正确**

**答案**

随机快速排序与快速排序的区别是随机快速排序随机选择基准元素，而快速排序的确定性算法选择固定位置的元素作为基准元素。

随机快速排序的时间复杂度趋于O(nlogn)。

以下随机化算法能得能保证得到的解是正确解的算法是（）

A. 蒙特卡罗算法

B. 拉斯维加斯算法

C. 数值随机化算法

D. 舍伍德算法

**回答正确**

**答案**

拉斯维加斯算法

舍伍德算法

随机化算法的特征是对所求解问题的同一实例用同一随机化算法求解两次可能得到完全不同的效果，这两次求解问题所需的时间甚至所得到的结果一定会有相当大的差别。

正确错误

**回答正确**

**答案**

错误

蒙特卡罗算法用于求问题的正确解。

正确错误

**回答正确**

**答案**

错误

拉斯维加斯算法绝不返回错误的解，但有时得不到问题的解。可以通过多次执行提高算法得到解的概率。

正确错误

**回答正确**

**答案**

正确

随机快速排序随机选择一个元素作为基准元素，根据划分的结果，递归解决两个子问题。

正确错误

**回答正确**

**答案**

正确

第六周作业

总分：100分

得分：95.0分

以下算法框架中，哪个是排列树模型的算法设计模式（）

A. def Backtrack (t):     if (t>n):        output(x)     else:       for i in range(1,m+1):          if (constraint(t) and bound(t)):                x[t]=i                做其他相关标识                Backtrack(t+1)                做其他相关标识的反操作

B. def Backtrack (t):     if (t>n):          output(x)     else:          for i in range(t,n+1):           x[t], x[i]←x[i], x[t]           if (constraint(t) and bound(t)):                  Backtrack(t+1)           x[t], x[i]←x[i], x[t]

C. def Backtrack (int t):    if (t>=n):         output(x)    else:       for  i  in range(s(n,t),e(n,t)):           x[t]=d(i)           if (constraint(t) and bound(t)):               Backtrack(t+1)

D. def Backtrack (int t):   if (t>n):       output(x)   if(constraint(t)):       做相关标识      Backtrack(t+1)      做相关标识的反操作   if(bound(t)):     做相关标识     Backtrack(t+1)     做相关标识的反操作

**回答正确**

**答案**

def Backtrack (t):     if (t>n):          output(x)     else:          for i in range(t,n+1):           x[t], x[i]←x[i], x[t]           if (constraint(t) and bound(t)):                  Backtrack(t+1)           x[t], x[i]←x[i], x[t]

最优化问题优化目标是使求目标函数最大化，基于回溯法求解该问题。如果对于解空间的任何分支X，均可求出目标函数值的两个上界lb1(X)和lb2(X)，且总有lb1（X）>=lb2(X),则如果想用于剪枝，从减少搜索节点的角度，哪个界限更优？

A. lb1

B. lb2

C. 二者等价

D. 依赖于具体输入

**回答正确**

**答案**

lb2

0-1背包问题的解空间结构属于（）

A. 排列树

B. 子集树

C. 满n叉树

D. 隐式图

**回答正确**

**答案**

子集树

以下关于回溯法的说法，错误的是

A. 回溯法一般会将解空间组织成树形结构并按照深度优先的顺序遍历

B. 回溯法可以适用于求所有解、某个解、最优解等各种问题

C. 回溯法能够保证生成时间复杂度较低的算法

D. 回溯法的编程中，有“当前搜索路径”的概念，需要保存当前路径上节点的状态

**回答正确**

**答案**

回溯法能够保证生成时间复杂度较低的算法

现有一个用于求解最优化问题的回溯算法，在搜索过程中涉及的函数的描述，错误的是

A. 违反约束函数的分支不属于问题的定义域

B. 违反限界函数的分支不需要访问，不能够得到更优解

C. 目标函数是衡量解的优劣程度的函数

D. 在目标函数最小化问题中，限界函数应当使用上界

**回答正确**

**答案**

在目标函数最小化问题中，限界函数应当使用上界

关于旅行商问题的说法，错误的是

A. 旅行商问题的解空间与最短路径问题相同

B. 旅行商问题的优化目标是回路长度最短

C. 有4个点的旅行商问题的两个回路，ABCDA和BCDAB，实际上是两个相同的回路

D. 旅行商问题无法用穷举求解，因为回路数目太多

**回答正确**

**答案**

旅行商问题的解空间与最短路径问题相同

以下有关旅行商问题的递归代码，根据注释判断空缺部分填写正确的是（）。 def Traveling(t): ( )#到达叶子结点     #g存储图的邻接矩阵，x是存储解向量，初始化为x[1:n]={1,2,...,n},cl是当前已走的路经长度，bestl是当前已找到的最短路径长度。       if(g[x[n] ][1] !=∞ and (cl+g[x[n]][1]< bestl ) ):          for j in range(1,n+1):                  bestx[j]=x[j]          bestl=cl+g[x[n]][1] else:#没有到达叶子结点        ( ) #控制当前节点的分支数目，即对xt的所有可能的取值。     if( g[x[t-1]][x[j]] !=∞ and(cl+g[x[t-1]][x[j]]< bestl) ): #保存第t个要去的城市编号到x[t]中，进入到第t+1层            x[t],x[j]=x[j],x[t] #交换两个元素的值              cl+=g[x[t-1]][x[j]]               Traveling(t+1)#从第t+1层的扩展结点继续搜索               #第t+1层搜索完毕，回溯到第t层             cl-=g[x[t-1]][x[j]]             x[t],x[j]=x[j],x[t]

A. 空1：if(t==n) 空2：for(j=t;j<=n;j++)

B. 空1：if(t>n-1) 空2：for(j=1;j<=n;j++)

C. 空1：if(t>n) 空2：for(j=t;j<=n;j++)

D. 空1：if(t>=n-1) 空2：for(j=1;j<=n;j++)

**回答正确**

**答案**

空1：if(t>n) 空2：for(j=t;j<=n;j++)

有关回溯法说法正确的是（）

A. 回溯法是一种深度优先搜索的搜索算法

B. 回溯法是一种“能进则进、进不了则换、换不了则退（回溯）”的搜索方法

C. 回溯法是一种宽（广）度优先搜索的搜索算法

D. 回溯法是一种最大效益或最小费用优先搜索的方法

**回答正确**

**答案**

回溯法是一种深度优先搜索的搜索算法

回溯法是一种“能进则进、进不了则换、换不了则退（回溯）”的搜索方法

有关n皇后问题说法正确的是（）

A. 该问题的解的形式为(x1, x2, … , xn)，xi表示第i个皇后位于第i行、第xi列(i=1,2,3,...n)

B. 该问题的初始状态为：(0,0,...,0)

C. 该问题的解空间的组织结构可以是排列树，也可以是满n叉树。

D. 该问题只需要设置约束条件，不需要限界条件。 E、 该问题解向量中的任意两个分量xi,xj满足：xi≠xj且|i-j|≠|xi-xj|

**回答正确**

**答案**

该问题的解的形式为(x1, x2, … , xn)，xi表示第i个皇后位于第i行、第xi列(i=1,2,3,...n)

该问题的初始状态为：(0,0,...,0)

该问题的解空间的组织结构可以是排列树，也可以是满n叉树。

该问题只需要设置约束条件，不需要限界条件。 E、 该问题解向量中的任意两个分量xi,xj满足：xi≠xj且|i-j|≠|xi-xj|

下述有关搜索过程描述错误的是（）

A. 当解空间结构是一棵树时，搜索从根开始

B. 搜索过程中，正在生成孩子的节点称为扩展节点

C. 搜索过程中，所有孩子节点均已生成的节点称为扩展节点

D. 搜索过程中，所有孩子节点均已生成的节点称为活结点节点 E、 搜索过程中，所有孩子节点均已生成的节点称为死节点 F、 搜索过程动态生成的树称为搜索树

**回答正确**

**答案**

搜索过程中，所有孩子节点均已生成的节点称为扩展节点

搜索过程中，所有孩子节点均已生成的节点称为活结点节点 E、 搜索过程中，所有孩子节点均已生成的节点称为死节点 F、 搜索过程动态生成的树称为搜索树

以下描述中，影响回溯法的搜索效率的是（）

A. 问题的解空间，即搜索范围

B. 设定的约束函数和限界函数

C. 搜索方法

D. 满足约束条件和限界条件的节点数目

**回答正确**

**答案**

问题的解空间，即搜索范围

设定的约束函数和限界函数

满足约束条件和限界条件的节点数目

以下有关子集树的描述中说法正确的是（）

A. 当所给的问题是从n个元素组成的集合S中找出满足某种性质的一个子集时，相应的解空间树称为子集树。

B. 子集树模型解的形式为n元组（x1,x2,…,xn），分量xi(i=1,2,…,n)表示第i个元素是否在子集中。

C. 子集树模型的解向量中，分量xi的取值为0或1，xi=0表示第i个元素不在子集中；xi=1表示第i个元素在子集中。

D. 旅行售货员问题可以开用子集树模型求解 E、 最优装载问题可以采用子集树模型求解 F、 0-1背包问题可以采用子集树模型求解

**回答正确**

**答案**

当所给的问题是从n个元素组成的集合S中找出满足某种性质的一个子集时，相应的解空间树称为子集树。

子集树模型解的形式为n元组（x1,x2,…,xn），分量xi(i=1,2,…,n)表示第i个元素是否在子集中。

子集树模型的解向量中，分量xi的取值为0或1，xi=0表示第i个元素不在子集中；xi=1表示第i个元素在子集中。

有关子集树描述中，说法错误的是（）

A. 子集树的根结点为问题的初始状态

B. 子集树的中间结点为搜索过程中形成的某中间状态

C. 子集树的叶子结点为问题结束状态

D. 子集树的分支表示从一个状态过渡到另一个状态的行为 E、 子集树中从根结点到叶子结点的路径是一个可行解（一个子集） F、 子集树的深度等于问题的规模加1

**回答正确**

钟表的特写

低可信度描述已自动生成**答案**

子集树的根结点为问题的初始状态

子集树的中间结点为搜索过程中形成的某中间状态

子集树的叶子结点为问题结束状态

子集树的分支表示从一个状态过渡到另一个状态的行为 E、 子集树中从根结点到叶子结点的路径是一个可行解（一个子集） F、 子集树的深度等于问题的规模加1

有关0-1背包问题说法正确的是（）

A. 该问题的解的形式为(x1, x2, … , xn)，xi(i=1,2,3,...n)的取值为0或1

B. 该问题的解空间的组织结构可以是排列树。

C. 该问题需要设置约束条件，也可以设置限界条件。

D. 该问题只需要设置约束条件，不需要限界条件。

**回答正确**

**答案**

该问题的解的形式为(x1, x2, … , xn)，xi(i=1,2,3,...n)的取值为0或1

该问题需要设置约束条件，也可以设置限界条件。

有关下图说法正确的是（）

A. 该树表示的问题的规模为3

B. 该树为一棵排列树

C. 该树表示的问题规模为4。

D. 该树为一棵子集树

**回答正确**

**答案**

该树表示的问题的规模为3

该树为一棵排列树

有关批处理作业调度问题说法正确的是（）

A. 该问题的解形式为(x1,x2,…,xn)，xi取值范围为：令S={1,2,…,n}，则xi∈S-{x1,x2,…,xi-1}，i=1,2,...,n

B. 该问题的解空间的组织结构是排列树。

C. 该问题需要设置约束条件，不需要限界条件。

D. 该问题不需要设置约束条件，只需要限界条件。 E、 该问题既需要设置约束条件，也需要限界条件。

**回答正确**

**答案**

该问题的解形式为(x1,x2,…,xn)，xi取值范围为：令S={1,2,…,n}，则xi∈S-{x1,x2,…,xi-1}，i=1,2,...,n

该问题的解空间的组织结构是排列树。

该问题不需要设置约束条件，只需要限界条件。 E、 该问题既需要设置约束条件，也需要限界条件。

有关旅行售货员问题说法正确的是（）

A. 该问题的解形式为(x1,x2,…,xn)，xi取值范围为：令S={1,2,…,n}，则xi∈S-{x1,x2,…,xi-1}

B. 该问题的解空间的组织结构是排列树。

C. 该问题需要设置约束条件，不需要限界条件。

D. 该问题不需要设置约束条件，只需要限界条件。 E、 该问题既需要设置约束条件，也可以设置限界条件。

**回答正确**

**答案**

该问题的解空间的组织结构是排列树。

+ 0.0 分

18 . 容易 （5分）

有关回溯法说法正确的是（）

A. 回溯法是一种深度优先搜索的搜索算法

B. 回溯法是一种“能进则进、进不了则换、换不了则退（回溯）”的搜索方法

C. 回溯法是一种宽（广）度优先搜索的搜索算法

D. 回溯法是一种最大效益或最小费用优先搜索的方法

 回答错误

**答案**

回溯法是一种深度优先搜索的搜索算法

回溯法是一种“能进则进、进不了则换、换不了则退（回溯）”的搜索方法

有关n皇后问题说法正确的是（）

A. 该问题的解的形式为(x1, x2, … , xn)，xi表示第i个皇后位于第i行、第xi列(i=1,2,3,...n)

B. 该问题的初始状态为：(0,0,...,0)

C. 该问题的解空间的组织结构可以是排列树，也可以是满n叉树。

D. 该问题只需要设置约束条件，不需要限界条件。

E. 该问题解向量中的任意两个分量xi,xj满足：xi≠xj且|i-j|≠|xi-xj|

**回答正确**

**答案**

该问题的解的形式为(x1, x2, … , xn)，xi表示第i个皇后位于第i行、第xi列(i=1,2,3,...n)

该问题的初始状态为：(0,0,...,0)

该问题的解空间的组织结构可以是排列树，也可以是满n叉树。

该问题只需要设置约束条件，不需要限界条件。

该问题解向量中的任意两个分量xi,xj满足：xi≠xj且|i-j|≠|xi-xj|

以下描述中，影响回溯法的搜索效率的是（）

A. 问题的解空间，即搜索范围

B. 设定的约束函数和限界函数

C. 搜索方法

D. 满足约束条件和限界条件的节点数目

**回答正确**

**答案**

问题的解空间，即搜索范围

设定的约束函数和限界函数

满足约束条件和限界条件的节点数目

第四周作业

总分：100分

得分：100.0分

解决给定的5个矩阵连乘问题：矩阵A1（3×2）、A2（2×5）、A3（5×10）、A4（10×2）和A5（2×3），设m[i][j]表示Ai...Aj的最优计算次序对应的乘法计算次数（最优值），P为存储矩阵行列的数组，其中P[i]是第i个矩阵的列、第i-1个矩阵的行。求解最优值递归关系是为：图示, 示意图

描述已自动生成，根据该递归关系式，求解过程中得到下面最优决策的二维表： 图表, 散点图

描述已自动生成由此，可得上述5个矩阵连乘的最优计算次序为（）

A. （A1（A2（A3（A4A5））））

B. （（A1A2）（A3（A4A5）））

C. （（A1A2）（（A3A4）A5））

D. （A1（（A2（A3A4））A5））

**回答正确**

**答案**

（A1（（A2（A3A4））A5））

关于动态规划和回溯法的区别，以下表述不正确的是

A. 动态规划和回溯法都可以用来求解最优化问题，但回溯法是基于枚举解的思想，动态规划则是基于构造子问题最优值关系的方式

B. 在遇到重叠子问题的时候，动态规划思想会使用存储最优值的方式直接排除，而回溯法一般做法是设法避环和剪枝，降低其影响

C. 在求解相同问题时，动态规划必然比回溯法浪费空间，但是更节约时间

**回答正确**

**答案**

在求解相同问题时，动态规划必然比回溯法浪费空间，但是更节约时间

关于动态规划与分治法的区别，表述不正确的是

A. 动态规划划分的子问题一般具有重叠子问题，分治法则通常互不相交

B. 动态规划建立在描述子问题最优值关系的状态转移方程基础上，分治法一般不需要建立类似的最优值之间的数量关系

C. 分治法能写成递归形式，动态规划不能写成递归形式

D. 动态规划一般用来求解最优化问题，分治法多不用于求解最优化问题，

**回答正确**

**答案**

分治法能写成递归形式，动态规划不能写成递归形式

矩阵连乘问题中，A1矩阵大小是100\*5,A2矩阵大小为5\*30，A3矩阵大小为30\*10，则乘法次序 (A1\*A2)\*A3需要的乘法次数是

A. 15000

B. 30000

C. 45000

D. 450000000

**回答正确**

**答案**

45000

规模为5矩阵连乘问题,计算次序有()种。

A. 10

B. 12

C. 14

D. 16

**回答正确**

**答案**

14

根据下面斐波那契数列的递归算法，可知斐波那契数列的第n项的递归式为（）。 def Fibonacci(int num):  if(num == 0 || num == 1):        return num    return  Fibonacci(num-1)+Fibonacci(num - 2)

A. Fibonacci(n)=0  当n=0时

B. Fibonacci(n)=1  当n=1时

C. Fibonacci(n)=Fibonacci(n-1)+Fibonacci(n-2) 当n〉1时

D. Fibonacci(n)=Fibonacci(n-2)+Fibonacci(n-3) 当n〉1时

**回答正确**

**答案**

Fibonacci(n)=Fibonacci(n-1)+Fibonacci(n-2) 当n〉1时

下面代码为求n！的递归算法，该代码反应的n!问题递归实现的停止条件（边界条件）为（）。

def fun(n):

  if (n == 1):

    return 1

else :

  return fun(n - 1) \* n

A. n!=1  当n=0时

B. n!=1  当n=1时

C. n!=1  当n〈1时

D. n!=1  当n〈=1时

**回答正确**

**答案**

n!=1  当n=1时

合并排序的空间复杂度为()

A. θ(logn)

B. θ(n)

C. θ(nlogn)

D. θ(n\*n)

**回答正确**

**答案**

θ(n)

以下哪个问题的时间复杂度与输入序列有关（）

A. 二分查找

B. 最小值问题

C. 合并排序

D. 以上都不对

**回答正确**

**答案**

二分查找

以下函数的功能是（） def Q(R, low，high):   if(low<high):#仅当区间长度大于1时才须排序      pivotpos=Partition(R,low,high) #划分后的基准元素所对应的位置       Q(R,low,pivotpos-1)#对左区间递归排序       Q(R,pivotpos+1,high)#对右区间递归排序

A. 二分查找

B. 二分求最值

C. 合并排序

D. 快速排序

**回答正确**

**答案**

快速排序

以下代码功能为合并排序，请根据注释按照数顺序选择合适的语句填入对应的括号。

def MergeSort(A， low， high):

  if (low < high):

    （）#分解

（）# 递归序列左半部分

（）# 递归序列右半部分

Merge(A, low, middle, high)# 子问题的解合并成原问题的解

A. middle=(high-low)/2; MergeSort(A,low,middle); MergeSort(A,middle+1,high);

B. middle=(low+high)/2; MergeSort(A,low,middle); MergeSort(A,middle+1,high);

C. middle=(low+high)/2; MergeSort(A,middle+1,high);  MergeSort(A,low,middle);

D. middle=(high-low)/2; MergeSort(A,middle+1,high);  MergeSort(A,low,middle);

**回答正确**

**答案**

middle=(low+high)/2; MergeSort(A,low,middle); MergeSort(A,middle+1,high);



合并排序的分治算法时间复杂度的是()

A. O(logn)

B. O(nlogn)

C.

D.

**回答正确**

**答案**

O(nlogn)

矩阵连乘问题中有多个矩阵相乘，问题是安排矩阵相乘的先后顺序，使总乘法次数最少，例如 有[A][B]C三个矩阵，则可行的顺序有ABC\ACB\CAB\CBA\BAC\BCA六个。

正确错误

**回答正确**

**答案**

错误

以动态规划求解0-1背包问题，背包容量可以是任意实数。

正确错误

**回答正确**

**答案**

错误

有关矩阵连乘问题说法正确的是()

A. 矩阵A i...A j连乘,其中A k的行列为(p k×q k),k=i,i+1,...,j,其结果矩阵的行列为(p i×q j)。

B. n个矩阵连乘A 1...A n,其子问题为A i...A j连乘,1≤i≤j≤n,其中i=j表示规模为1的子问题,其需要的乘法次数为0。

C. 设矩阵A i...A j连乘最少的乘法次数为c[i][j],矩阵A i...A j连乘的子问题为矩阵A i...A k连乘和矩阵A k+1...A j连乘,则最优值的递归关系式表示为c[i][j]=c[i][k]+c[k+1][j]+p iq jq k

D. 矩阵连乘问题的时间复杂度为O(n 2)

**回答正确**

**答案**

矩阵A i...A j连乘,其中A k的行列为(p k×q k),k=i,i+1,...,j,其结果矩阵的行列为(p i×q j)。

n个矩阵连乘A 1...A n,其子问题为A i...A j连乘,1≤i≤j≤n,其中i=j表示规模为1的子问题,其需要的乘法次数为0。

动态规划的基本要素是()

A. 重叠子问题

B. 最优子结构性质

C. 自底向上的求解方式

D. 自顶向下的递归求解方式

**回答正确**

**答案**

重叠子问题

最优子结构性质

自底向上的求解方式

有关动态规划描述正确的是()

A. 动态规划将多阶段决策问题转化为单阶段决策问题。

B. 动态规划往往用于求解某种最优性质的问题。

C. 适用动态规划求解的问题经分解得到的各个子问题往往不是相互独立的。

D. 动态规划求解时往往采用填表的方法记录问题最优值。 E、 动态规划划分的各子问题与原问题相同,一般递归求解子问题。 F、 动态规划求解某种最优性质的问题时,整体的最优值和子问题的最优值之间存在递归关系。

**回答正确**

**答案**

动态规划将多阶段决策问题转化为单阶段决策问题。

动态规划往往用于求解某种最优性质的问题。

适用动态规划求解的问题经分解得到的各个子问题往往不是相互独立的。

动态规划求解时往往采用填表的方法记录问题最优值。 E、 动态规划划分的各子问题与原问题相同,一般递归求解子问题。 F、 动态规划求解某种最优性质的问题时,整体的最优值和子问题的最优值之间存在递归关系。

设c[i][j]表示序列Xi和Yj的最长公共子序列的长度。则它的递推关系式为： 文本

中度可信度描述已自动生成则，根据给定的X=={A, B, C, B, D, A, B}和Y={B, D, C, A, B, A}从上到下填写缺失值。 图片包含 图表

描述已自动生成208089516230602074545140081017633951191260

A. 2 3 3

B. 2 2 2

C. 3 4 4

D. 3 3 3

**回答正确**

**答案**

3 4 4

给定序列X={A, B, C, B, D, A, B}和Y={B, D, C, A, B, A}，它们的最长公共子序列是（）。

A. BCBA

B. BCDA

C. BDAB

D. BCAA

**回答正确**

**答案**

BCBA

按照顺序排列动态规划的求解步骤,正确的是() （1)递归定义最优值。 （2)以自底向上的方式计算出最优值,并记录相关信息。 （3）分析最优解子结构性质。 （4)构造出最优解。

A. (1),(2),(3),(4)

B. (1),(3),(2),(4)

C. (3),(1),(2),(4)

D. (1),(2),(4),(3)

**回答正确**

**答案**

(3),(1),(2),(4)

2022-3-13-第三周作业

总分：100分

得分：100.0分

以下问题中，哪些问题的分治算法消耗的时间与输入序列无关.（）

A. 二分查找

B. 合并排序

C. 快速排序

D. 最小值问题

**回答正确**

**答案**

合并排序

最小值问题

有关2个n位大整数乘法问题说法正确的是（）。

A. 将两个n位大整数分解为4个规模大致相等的n/2位整数的整数乘法问题

B. 递归解决4个子问题

C. 子问题的解需要归并成原问题的解

D. 子问题的解本身就是原问题的解

**回答正确**

**答案**

将两个n位大整数分解为4个规模大致相等的n/2位整数的整数乘法问题

递归解决4个子问题

子问题的解需要归并成原问题的解

分治算法的步骤有（）。

A. 分解

B. 治理

C. 递归

D. 合并

**回答正确**

**答案**

分解

治理

分治算法的思想是（）。

A. 将规模较大的问题划分为规模较小的相同子问题

B. 子问题之间相互独立

C. 子问题之间不相互独立

D. 递归解决划分得到的子问题 E、 将子问题的解归并得到原问题的解

**回答正确**

**答案**

将规模较大的问题划分为规模较小的相同子问题

子问题之间相互独立

递归解决划分得到的子问题 E、 将子问题的解归并得到原问题的解

大整数A和B的乘法,将A分成位数大致相等的两部分A1和A2 ，将B分成位数大致相等的两部分B1和B2,以下描述正确的是()。

A. 子问题的解归并为原问题解的方法为:A×B=10nA1B1+10n/2(A1B2+A2B1)+A2B2

B. 子问题的解归并为原问题解的方法为:A×B=10nA1B1+10n/2((A1-A2)(B2-B1)+A1B1+A2B2)+A2B2

C. 子问题的解归并为原问题解的方法为:A×B=10nA1B1+10n/2((A1+A2)(B1+B2)-A1B1-A2B2)+A2B2

D. 以上方法都不对。

**回答正确**

**答案**

子问题的解归并为原问题解的方法为:A×B=10nA1B1+10n/2(A1B2+A2B1)+A2B2

子问题的解归并为原问题解的方法为:A×B=10nA1B1+10n/2((A1-A2)(B2-B1)+A1B1+A2B2)+A2B2

子问题的解归并为原问题解的方法为:A×B=10nA1B1+10n/2((A1+A2)(B1+B2)-A1B1-A2B2)+A2B2

关于快速排序分治算法时间复杂度描述正确的是()

A. 快速排序分治算法最好情况下的时间复杂度为O(nlogn).

B. 快速排序分治算法最坏情况下的时间复杂度为O(n 2).

C. 快速排序分治算法平均情况下的时间复杂度为O(n 2).

D. 二快速排序分治算法平均情况下的时间复杂度为O(nlogn).

**回答正确**

**答案**

快速排序分治算法最好情况下的时间复杂度为O(nlogn).

快速排序分治算法最坏情况下的时间复杂度为O(n 2).

二快速排序分治算法平均情况下的时间复杂度为O(nlogn).

有关快速排序的分治算法描述正确的是()。

A. 快速排序A[left,right],选取基准元素的方法,将待排序元素分解为两个子问题。

B. 快速排序基准元素的选取可以是待排序元素中的任何一个元素。

C. 快速排序划分的两个子问题规模大致相等。

D. 快速排序A[left,right],递归算法的边界条件是left≥right

**回答正确**

**答案**

快速排序A[left,right],选取基准元素的方法,将待排序元素分解为两个子问题。

快速排序基准元素的选取可以是待排序元素中的任何一个元素。

快速排序A[left,right],递归算法的边界条件是left≥right

关于二分查找时间复杂度描述正确的是()

A. 二分查找算法最好情况下的时间复杂度为O(1).

B. 二分查找算法最坏情况下的时间复杂度为O(n).

C. 二分查找算法最坏情况下的时间复杂度为O(logn).

D. 二分查找算法平均情况下的时间复杂度为O(logn).

**回答正确**

**答案**

二分查找算法最好情况下的时间复杂度为O(1).

二分查找算法最坏情况下的时间复杂度为O(logn).

二分查找算法平均情况下的时间复杂度为O(logn).

有关合并排序的分治算法描述正确的是()

A. 合并排序A[left,right]的元素,采用的分解方法是(left+right)/2。

B. 合并排序A[left,right]的元素,采用的分解方法是(right-left)/2。

C. 合并排序A[left,right]的元素,需要治理规模大致等于(right-left+1)/2的两个子问题。

D. 合并排序需要将两个有序的子序列归并成一个有序的子序列。

**回答正确**

**答案**

合并排序A[left,right]的元素,采用的分解方法是(left+right)/2。

合并排序A[left,right]的元素,需要治理规模大致等于(right-left+1)/2的两个子问题。

合并排序需要将两个有序的子序列归并成一个有序的子序列。

有关循环赛日程表分治算法描述正确的是()

A. 循环赛日程表给定2 k个运动员,采用2 k/2的方法将运动员分成两组。

B. 循环赛日程表算法先安排组内的赛程,再安排两组对打。

C. 循环赛日程表算法的边界条件是两个运动员,一天的比赛。

D. 循环赛日程表算法为2 k个运动员安排了2 k-1天的比赛。

**回答正确**

**答案**

循环赛日程表给定2 k个运动员,采用2 k/2的方法将运动员分成两组。

循环赛日程表算法先安排组内的赛程,再安排两组对打。

循环赛日程表算法的边界条件是两个运动员,一天的比赛。

下述关于二分查找(折半查找)算法描述正确的是()

A. 二分查找是在任意给定的n个元素序列中查找指定元素。

B. 二分查找的序列为A[left,right],分解操作为:(right-left)/2

C. 二分查找根据比较的结果,好的情况是相等,算法结束。坏的情况是进入其中一个子问题继续查找。

D. 若二分查找的序列为A[left,right],用递归来解决子问题,则边界条件是left>right。

**回答正确**

**答案**

二分查找根据比较的结果,好的情况是相等,算法结束。坏的情况是进入其中一个子问题继续查找。

若二分查找的序列为A[left,right],用递归来解决子问题,则边界条件是left>right。

分治算法核心就是分而治之,其中的“治”描述正确的是()。

A. 分治法通过治理小问题来治理大问题。

B. 分治法递归治理小问题。

C. 分治法需要将子问题的解归并成大问题的解。

D. 治理子问题时,会有重复性治理子问题的现象。

**回答正确**

**答案**

分治法通过治理小问题来治理大问题。

分治法递归治理小问题。

分治法需要将子问题的解归并成大问题的解。

分治算法的基本思想描述正确的是()

A. 分治法将规模大的问题分解成规模较小的问题解决。

B. 分治法划分的小问题相互重叠。

C. 分治法一般采用递归的方法解决子问题。

D. 分治法划分的小问题规模小到一定程度时容易解决。

**回答正确**

**答案**

分治法将规模大的问题分解成规模较小的问题解决。

分治法一般采用递归的方法解决子问题。

分治法划分的小问题规模小到一定程度时容易解决。

根据下面斐波那契数列的递归算法，可知斐波那契数列的第n项的递归式为（）。 def Fibonacci(int num):  if(num == 0 || num == 1):        return num    return  Fibonacci(num-1)+Fibonacci(num - 2)

A. Fibonacci(n)=0  当n=0时

B. Fibonacci(n)=1  当n=1时

C. Fibonacci(n)=Fibonacci(n-1)+Fibonacci(n-2) 当n〉1时

D. Fibonacci(n)=Fibonacci(n-2)+Fibonacci(n-3) 当n〉1时

**回答正确**

**答案**

Fibonacci(n)=Fibonacci(n-1)+Fibonacci(n-2) 当n〉1时

下面代码为求n！的递归算法，该代码反应的n!问题递归实现的停止条件（边界条件）为（）。

def fun(n):

  if (n == 1):

    return 1

else :

  return fun(n - 1) \* n

A. n!=1  当n=0时

B. n!=1  当n=1时

C. n!=1  当n〈1时

D. n!=1  当n〈=1时

**回答正确**

**答案**

n!=1  当n=1时

以下哪个问题的时间复杂度与输入序列有关（）

A. 二分查找

B. 最小值问题

C. 合并排序

D. 以上都不对

**回答正确**

**答案**

二分查找

以下函数的功能是（） def Q(R, low，high):   if(low<high):#仅当区间长度大于1时才须排序      pivotpos=Partition(R,low,high) #划分后的基准元素所对应的位置       Q(R,low,pivotpos-1)#对左区间递归排序       Q(R,pivotpos+1,high)#对右区间递归排序

A. 二分查找

B. 二分求最值

C. 合并排序

D. 快速排序

**回答正确**

**答案**

快速排序

以下代码功能为合并排序，请根据注释按照数顺序选择合适的语句填入对应的括号。

def MergeSort(A， low， high):

  if (low < high):

    （）#分解

（）# 递归序列左半部分

（）# 递归序列右半部分

Merge(A, low, middle, high)# 子问题的解合并成原问题的解

A. middle=(high-low)/2; MergeSort(A,low,middle); MergeSort(A,middle+1,high);

B. middle=(low+high)/2; MergeSort(A,low,middle); MergeSort(A,middle+1,high);

C. middle=(low+high)/2; MergeSort(A,middle+1,high);  MergeSort(A,low,middle);

D. middle=(high-low)/2; MergeSort(A,middle+1,high);  MergeSort(A,low,middle);

**回答正确**

**答案**

middle=(low+high)/2; MergeSort(A,low,middle); MergeSort(A,middle+1,high);

棋盘覆盖问题的分解方法为（）。

A. 图示

中度可信度描述已自动生成

B. 

C. 

D. 以上分解的方法都不对

**回答正确**

**答案**



合并排序的分治算法时间复杂度的是()

A. O(logn)

B. O(nlogn)

C.

D. 

**回答正确**

**答案**

O(nlogn)

2022-3-4-第二周作业

总分：100分

得分：89.0分

求递推方程图示

中度可信度描述已自动生成得到的解是\_\_\_

**回答正确**

**答案**

O(nlogn)

求递推方程图示, 文本

中度可信度描述已自动生成得到的解是\_\_\_

**回答正确**

**答案**

O(logn)

+ 0.0 分

3 . 普通 （5分）

求文本, 信件

描述已自动生成递推方程的解是\_\_\_

 回答错误

**答案**

<img src="https://cdn1.qingline.net/38775d2de17d5c2e9a3628e30523adf1.png" />

求递推方程得到的解是（）

**回答正确**

**答案**

O(logn)



求递推方程的解是（）

**回答正确**

**答案**

O(n^2)

物品可以切割的背包问题的最佳贪心策略不一定能保证装入背包的物品总价值最大。

正确错误

手表的钟表

描述已自动生成 **回答正确**

**答案**

错误

+ 0.0 分

图片包含 图表

描述已自动生成7 . 普通 （5分）

对下图分别用Kruskal算法和Prim算法求最小生成树。

 回答错误

**答案**

+ 4.0 分

8 . 普通 （5分）

磁带最优存储问题：设有n 个程序{1,2,…, n }要存放在长度为L的磁带上。程序i存放在磁带上的长度是Li， 1≤i≤n。这n 个程序的读取概率分别是p1,p2,...,pn,且p1+p2+...+pn = 1。如果将这n 个程序按 1,2,....,n 的次序存放，则读取程序i所需的时间tr=c\*(P1×L1+P2×L2+...+Pr×Lr)。这n 个程序的平均读取时间为 t1+t2+...+tn。实际上第k个程序的读取概率为ak/(a1+a2+...+an)。对所有输入均假定c=1。磁带最优存储问题要求确定这n个程序在磁带上的一个存储次序，使平均读取时间达到最小。试设计一个解此问题的算法,并分析算法的正确性和计算复杂性。

**回答正确**

**答案**

def MinTime(n,l,p): t = [] a = 0 for i in range(n): a=a+p[i] #计算概率和 for i in range(n): t.append(l[i]\*p[i]/a)#存放i个程序的长度与读取概率的积 t.sort()#由小到大到排序 sum\_time=0 for i in range(n): sum\_time=sum\_time+(n-i) \* t[i] #计算n个程序的平均读取时间 sum\_time = n\*t[0]+(n-1)\*t[1]+...+t[n-1] return round(sum\_time,4)#四舍五入，保留4位小数 if \_\_name\_\_=="\_\_main\_\_": n = 5 l = [71,46,9,73,35] p = [872,452,265,120,85] print(MinTime(n,l,p)) 输出结果为： 85.6193

设字符m1，m2，…m10的查阅频率依次为：0.05，0.01，0.01，0.10，0.03，0.17，0.0 2，0.24，0.31，0.06。试构造对应的哈夫曼(Haffman)编码，并画出相应的编码树，同时写出ml,m2,…m10的编码。

**回答正确**

**答案**

图示

描述已自动生成每个字符的编码为从根节点到该字符所在叶子结点的路径上的0,1组成的串。

在10000个元素中找到前100个最大的元素，如果使用以下某个数据结构作为辅助，比较合适的是()

A. 堆

B. 并查集

C. 循环链表

D. 哈希表

**回答正确**

**答案**

堆

给定下面的有向、连通带权图 用dijkstra算法，找从源点1到其他各个顶点的最短路径。算法运行若干步以后，得到各数据结构的数据如下（数组下标从1开始，表示顶点编号）： 下标 11 22 3 4 5 6 7 88 S 1 1 0 1 0 1 1 0 dist 0 2 8 1 6 3 3 11 pre 0 1 2 1 7 1 4 7 根据当前状态，可判断从初始状态到当前状态已经做了（）次贪心选择。

A. 1

B. 2

C. 3

D. 4

**回答正确**

手表是蓝色的钟表

中度可信度描述已自动生成**答案**

4

用Prim算法求解上图的最小生成树，初始时，集合S={a}，集合V-S={b,c,d,e,f,g}，第六步贪心选择的边是（）。

A. （a,b）

B. （c,d）

C. （b,c）

D. （c,f）

**回答正确**

**答案**

钟表的特写

描述已自动生成（c,d）

+ 4.0 分

13 . 普通 （4分）

用Prim算法求解上图的最小生成树，初始时，集合S={a}，集合V-S={b,c,d,e,f,g}，第三步贪心选择的边是（）。

A. （a,b）

B. （b,c）

C. （c,d）

D. （c,f）

**回答正确**

**答案**

（c,f）

+ 4.0 分

14 . 容易 （4分）

有钟表的手表

中度可信度描述已自动生成用Prim算法求解上图的最小生成树，初始时，集合S={a}，集合V-S={b,c,d,e,f,g}，第一步贪心选择的边是（）。

墙上的钟表

低可信度描述已自动生成A. （a,b）

B. （b,c）

C. （c,d）

D. （c,f）

**回答正确**

**答案**

（a,b）

+ 4.0 分

15 . 容易 （4分）

手表是蓝色的钟表

中度可信度描述已自动生成用Kurskal算法求解上图的最小生成树，第一步贪心选择的边是（）。

A. （a,b）

B. （b,c）

C. （c,g）

D. （c,f）

**回答正确**

**答案**

（c,f）

+ 4.0 分

16 . 普通 （4分）

给定一个有向连通带权图G=(V,E),n个顶点,e条边,Dijsktra算法的时间复杂度为()

A. O(n 2)

B. O(n 3)

C. O(eloge)

D. O(nlogn)

**回答正确**

**答案**

O(n 2)

+ 4.0 分

17 . 容易 （4分）

背包问题: n个物品和1个背包。对物品i,其价值为vi,重量为wi,背包的容量为W。如何选取物品装入背包,使背包中所装入的物品的总价值最大?物品可以分割。该问题的贪心策略是()。

A. 重量小的优先装入背包

B. 体积小的优先装入背包

C. 价值大的优先装入背包

D. 单位重量的价值大的优先装入背包

**回答正确**

**答案**

单位重量的价值大的优先装入背包

+ 4.0 分

18 . 普通 （4分）

调度问题:有n个客户带来n项任务,每项加工时间已知,设为ti,i=1,2,…,n。从0时刻开始,陆续安排到一台机器上加工。每个任务的完成时间是从0时刻到该任务加工完成的时间。为了使尽可能多的客户满意,我们希望找到是的总等待时间最少的调度方案。该问题的贪心策略是()

A. 加工时间长的优先安排

B. 加工时间短的优先安排

C. 完成时间早的优先安排

D. 等待时间长的优先安排

**回答正确**

**答案**

加工时间短的优先安排

+ 4.0 分

19 . 容易 （4分）

找零钱问题的贪心策略是()

A. 面值大的钱币优先找出

B. 面值小的钱币优先找出

C. 面值小于待找钱数且面值最大的优先找出

D. 以上都不对

**回答正确**

**答案**

面值小于待找钱数且面值最大的优先找出

+ 4.0 分

20 . 容易 （4分）

物品不可拆开的最优装载问题的贪心策略是()

A. 体积大的集装箱优先装

B. 体积小的集装箱优先装

C. 重量大的集装箱优先装

D. 重量小的集装箱优先装

**回答正确**

**答案**

重量小的集装箱优先装

+ 4.0 分

21 . 容易 （4分）

会场安排问题的最好的贪心策略是()

A. 在不冲突的情况下，开始时间早的优先安排

B. 在不冲突的情况下，使用时间短的优先安排

C. 在不冲突的情况下，使用时间长的优先安排

D. 在不冲突的情况下，结束时间早的优先安排

**回答正确**

**答案**

在不冲突的情况下，结束时间早的优先安排

+ 4.0 分

22 . 容易 （4分）

调度问题的算法设计策略是（）

A. 加工时间短的优先安排

B. 加工时间长的优先安排

C. 等待时间短的优先安排

D. 以上都不对

**回答正确**

**答案**

加工时间短的优先安排

2022-2-23-第一周作业

总分：100分

得分：100.0分

+ 4.0 分

1 . 容易 （4分）

2 n=O(100n 2)

正确错误

**回答正确**

**答案**

错误

+ 3.0 分

2 . 容易 （3分）

10=θ(log10)

正确错误

**回答正确**

**答案**

正确

+ 3.0 分

3 . 容易 （3分）

2 n=O(3 n)

正确错误

**回答正确**

**答案**

正确

+ 3.0 分

4 . 容易 （3分）

logn 2=θ(logn+5)

正确错误

**回答正确**

**答案**

正确

+ 3.0 分

5 . 容易 （3分）

针对顺序查找算法，影响它时间复杂度的因素只有算法的输入序列()

正确错误

**回答正确**

**答案**

错误

+ 3.0 分

6 . 容易 （3分）

n!的时间复杂度为O(n)

正确错误

**回答正确**

**答案**

正确

+ 3.0 分

7 . 容易 （3分）

递归是指自己间接或直接调用自身

正确错误

**回答正确**

**答案**

正确

+ 3.0 分

8 . 普通 （3分）

算法的基本特征有（）

A. 输入

B. 输出

C. 有限性

D. 确定性

E. 可行性

**回答正确**

**答案**

输入

输出

有限性

确定性

可行性

+ 3.0 分

9 . 普通 （3分）

渐进复杂性的含义是（）情况下的复杂性。

A. 在最佳输入情况下

B. 问题规模趋向于无穷

C. 在最坏输入情况下

D. 平均各种输入之后

**回答正确**

**答案**

问题规模趋向于无穷

+ 3.0 分

10 . 普通 （3分）

n个连续自然数a1...an连加和问题算法（利用等差数列求和公式）的输入可以是什么（）。

A. a1，n

B. an,n

C. a1,an

D. a1,an,n

**回答正确**

**答案**

a1，n

an,n

a1,an

a1,an,n

+ 3.0 分

11 . 普通 （3分）

平均时间复杂度是指（）

A. 各种情况时间复杂度按概率的加权平均

B. 最好情况和最坏情况的时间复杂度的算术平均

C. 各种情况时间复杂度按概率的算术平均

D. 出现可能性最高的情况下的时间复杂度

**回答正确**

**答案**

各种情况时间复杂度按概率的加权平均

+ 3.0 分

12 . 容易 （3分）

算法的常见描述方式不包括（）

A. 代码

B. 甘特图

C. 伪代码

D. 流程图

**回答正确**

**答案**

甘特图

+ 3.0 分

13 . 容易 （3分）

算法的基本特性不包括（）

A. 先进性

B. 有穷性

C. 有输入输出

D. 无二义性

**回答正确**

**答案**

先进性

+ 3.0 分

14 . 普通 （3分）

阶乘问题求n!算法的时间复杂度为（）。

A. n

B. n!

C. 2n

D. n^2

**回答正确**

**答案**

n

+ 3.0 分

15 . 容易 （3分）

二分搜索（二分查找）算法的时间复杂度是（）。

A. n

B. logn

C. n^2

D. 2n

**回答正确**

**答案**

logn

+ 3.0 分

16 . 普通 （3分）

汉诺塔问题的时间复杂度是（）。

A. n!

B. 2^n

C. 2n

D. logn

**回答正确**

**答案**

2^n

+ 3.0 分

17 . 容易 （3分）

下述描述算法的方式采用的是算法的哪种描述方式（）？ 算法：gcd(m,n) 输入：非负整数m,n，其中m,n不全为0 输出：m与n的最大公约数 1.while m>0 do 2. r←n mod m 3. n ←m 4. m ←r 5.return n

A. 自然语言

B. 程序流程图

C. 伪码

D. 程序设计语言

**回答正确**

**答案**

伪码

+ 3.0 分

18 . 普通 （3分）

背包问题的算法设计策略是（）

A. 重量小的优先装

B. 价值大的优先装

C. 单位重量价值大的优先装

D. 以上都不对

**回答正确**

**答案**

单位重量价值大的优先装

+ 3.0 分

19 . 容易 （3分）

调度问题的算法设计策略是（）

A. 加工时间短的优先安排

B. 加工时间长的优先安排

C. 等待时间短的优先安排

D. 以上都不对

**回答正确**

**答案**

加工时间短的优先安排

+ 3.0 分

20 . 普通 （3分）

n个元素的冒泡排序代码如下：

def bubble\_sort(arr):

  for i in range(len(arr) - 1):

  for j in range(len(arr) - i - 1):

  if arr[j] > arr[j + 1]:

  arr[j], arr[j + 1] = arr[j + 1], arr[j]

return arr

请分析算法的时间复杂度，用O表示（）

A. O(1)

B. O(n)

C. 图示

低可信度描述已自动生成

D. O(nlogn)

**回答正确**

**答案**

图示

低可信度描述已自动生成

+ 3.0 分

21 . 普通 （3分）

百元买白鸡问题：鸡翁一，值钱五；鸡母一，值钱三；鸡雏三，值钱一；百钱买百鸡，则翁、母、雏各几何？设计一算法，则该算法的输入是（）

A. 100元

B. 100只鸡

C. 各种鸡的单价

D. 无需任何输入

**回答正确**

**答案**

无需任何输入

+ 3.0 分

22 . 普通 （3分）

下面算法最好情况下的时间复杂度\_\_\_，最坏情况下的时间复杂度为\_\_\_ def bubble\_sort(nums):         for i in range(len(nums) - 1):             swap\_flag = False  #改进后的冒泡，设置一个交换标志位             for j in range(len(nums) - i - 1):                 if nums[j]>nums[j+1]:                     nums[j],nums[j+1]=nums[j+1],nums[j]                     swap\_flag = True             if not swap\_flag:                 return nums  #若没有元素交换，则表示已经有序         return nums

**回答正确**

**答案**

O(n)、 <img src="https://cdn1.qingline.net/d61861a5b2faa1ecb1522d77d786eb71.png" />

+ 3.0 分

23 . 普通 （3分）

以下递归程序fun(5,0)输出的第一个元素是\_\_\_，求解过程中最大层次为\_\_\_ def fun(i,d):   if(i>1 and i%2!=0): fun(i-i//2,d+1)          if(i>1): fun(i//2,d+1)

**回答正确**

**答案**

1、4

+ 3.0 分

24 . 普通 （3分）

斐波那契数列的第1项为1，第2项为2，以后每一项等于前面两项之和，则第6项为\_\_\_

**回答正确**

**答案**

13

+ 3.0 分

25 . 普通 （3分）

冒泡排序时间复杂度是\_\_\_，堆排序时间复杂度是\_\_\_。

**回答正确**

**答案**

<img src="https://cdn1.qingline.net/8d1b0cce193d148418e59fa76b0dfd6c.png" />、nlogn

+ 3.0 分

26 . 容易 （3分）

递归算法必须具备的两个条件是\_\_\_和\_\_\_

**回答正确**

**答案**

边界条件或停止条件、递推方程或递归方程

+ 3.0 分

27 . 普通 （3分）

用O表示21+1/n的阶(O(1))

**回答正确**

**答案**

O(1)

+ 3.0 分

28 . 普通 （3分）

用O表示10log3 n的阶()

**回答正确**

**答案**

O(n)

+ 3.0 分

29 . 普通 （3分）

用O表示logn 3的阶()

**回答正确**

**答案**

O(logn)

+ 3.0 分

30 . 容易 （3分）

用O表示图示

描述已自动生成的阶\_\_\_

**回答正确**

**答案**

<img src="https://cdn1.qingline.net/46581b7980051edceb52b968f53e2e5d.png" />

+ 3.0 分

31 . 容易 （3分）

用O表示3n2+10n的阶\_\_\_

**回答正确**

**答案**

<img src="https://cdn1.qingline.net/fe158b0aad34f26a17d39bf06ec36749.png" />

+ 3.0 分

32 . 普通 （3分）

请分析选择排序算法的时间复杂度，用O表示为:\_\_\_ def selection\_sort(arr):     # 第一层for表示循环选择的遍数     for i in range(len(arr) - 1):           # 将起始元素设为最小元素         min\_index = i           # 第二层for表示最小元素和后面的元素逐个比较         for j in range(i + 1, len(arr)):               if arr[j] < arr[min\_index]:                  # 如果当前元素比最小元素小，则把当前元素角标记为最小元素角标                 min\_index = j          # 查找一遍后将最小元素与起始元素互换         arr[min\_index] , arr[i] = arr[i] , arr[min\_index]     return arr

**回答正确**

**答案**

<img src="https://cdn1.qingline.net/b9a1101afb3f1b7f1d08ba5e740c0590.png" />

+ 3.0 分

33 . 普通 （3分）

请分析下面算法的时间复杂度： def bubble\_sort(nums):         for i in range(len(nums) - 1):             swap\_flag = False  #改进后的冒泡，设置一个交换标志位             for j in range(len(nums) - i - 1):                 if nums[j]>nums[j+1]:                     nums[j],nums[j+1]=nums[j+1],nums[j]                     swap\_flag = True             if not swap\_flag:                 return nums  #若没有元素交换，则表示已经有序         return nums

**回答正确**

**答案**

最好情况下，比较n-1次，时间复杂度为O(n) 最坏情况下，比较n(n-1)/2次,时间复杂度为O(n^2)