2023学硕算法期末考试题

1、简答算法的概念和所具有的属性，算法复杂性有哪两种？给出算法时间复杂性在渐近意义下的阶的四中记号的含义。

算法：算法是指解决问题的一种方法或一个过程，更严格的来讲，算法是由若干条指令组成的有穷序列

算法具有的属性：输入、输出、确定性、有限性

算法复杂度：时间复杂度、空间复杂度

剩余内容参考课本

2、分别简述分治法、动态规划法与贪心算法的基本思想，并指出它们的相同点与不同点。

分治法的基本思想：将一个规模为 n 的问题分解为 k 个规模较小的子问题，这些子问题互相独立且与原问题相同，递归的求解这些子问题，然后将各个子问题的解合并得到原问题的解。

动态规划算法的基本思想：将待求解的问题分解成若干子问题，先求解子问题，再结合这些子问题的解得到原问题的解。按顺序求解子阶段，前一子问题的解，为后一子问题的解提供信息，无需重复计算。求解任何一个子问题时，列出各种可能的局部解，通过决策保留那些有可能达到的最优局部解，依次解决子问题，最后一个子问题就是初始问题的解

贪心算法：问题求解时，做出当前看来是最好的选择，并不从整体最优上进行考虑，总是选择当前看起来是最好的解。

动态规划与分治算法：

动态规划与分治法类似，基本思想都是将待求解问题分解成若干子问题，先求解子问题，再结合这些子问题得到原问题的解。

与分治法不同，适合用动态规划求解的问题分解得到的子问题往往不是互相独立的。

动态规划与贪心算法：

相同点：都要求有最优子结构性质。

不同点：动态规划要求子问题重叠，贪心算法要求具有贪心选择性质。

不同点：动态规划要求子问题重叠，贪心算法要求具有贪心选择性质。

3、由递归关系求通项，

考试题和下面的递推关系不一样，但是也要求实现递归和非递归代码

3.(10 分)已知 Fihonaci 数列$F(n)=\frac{1}{\sqrt{5}}[\left(\frac{1+\sqrt{5}}{2}\right)^{n+1}-\left(\frac{1-\sqrt{5}}{2}\right)^{n+1}]$，n=0,1,2.....。记一个新的数列 g(n)的递归表达式为：g(n)= $\begin{cases} g(n)=g(n-1)+g(n-2) \\ g(0)=g(1)=a \end{cases}$ n=2,3...。其中，a 是常数。试求出 g(n) 的非递归表达式。然后写出 g(n)的递归函数和非递归函数形式的算法实现伪代码。

根据特征方程：$x^2-x-1=0$

得出解：$\alpha=\frac{1+\sqrt{5}}{2},\beta=\frac{1-\sqrt{5}}{2}$

$g(0)=A+B=a$

$g(1)=A\alpha+B\beta=a$

得出$g(n)=\frac{\alpha^n(1+\sqrt{5})}{2\sqrt{5}}a-\frac{\beta^n(1-\sqrt{5})}{2\sqrt{5}}a=\frac{a}{\sqrt{5}}(\left(\frac{1+\sqrt{5}}{2}\right)^{n+1}-\left(\frac{1-\sqrt{5}}{2}\right)^{n+1})$

递归代码

```Python
def g(n):
    if n>1:
        return g(n-1)+g(n-2)
    else:
        return a
```

非递归

```Python
def g(n):
    g1= a
    g2 = a
    for i in range(2,n):
        g1 = g1+g2
```

```
        g2 = g1-g2
return g1
```

4、写出快速排序算法对一组数字排序的调用过程

实例略，归并排序过程也要看，可能会考

5、最长公共子序列画表格

6、单源最短路径 Dijkstra 算法画表

7.分别用贪心算法、动态规划算法、回溯算法、分支限界法设计 0–1 背包问题。要求：说明使用的算法策略；学出算法实验的主要步骤；分析算法的时间。

1、贪心算法：

策略：每一次选取单位价值高的物品放入背包。

基本步骤：首先计算每种物品的单位价值 vi/wi;然后依照贪心选择策略，将尽可能多的单位重量价值高的物品装入背包。若将这种物品全部装入背包后，背包内的物品总重量未超过 c，则选择单位重量价值次高的物品装入背包，以此策略，一直进行下去。

算法的时间：主要计算时间在于，将各种物品按照其单位重量的价值从大到小排序，因此，算法的时间上界为 O(nlogn)

2、动态规划算法

策略：建立最优值为 m(i,j)，即 m(i,j)是背包容量为 j，可以选择物品 I,i+1,….n 的递归表达式。

策略：建立最优值为 m(i,j)，即 m(i,j)是背包容量为 j，可以选择物品 I,i+1,....n 的递归表达式。

$$m(i,j)\begin{cases}\max\{m(i+1,j),m(i+1,j-wi)+vi\} & j \ge wi \\ m(i+1,j) & 0 \le j < wi\end{cases}$$

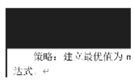
$$m(n,j)\begin{cases}vn & j \ge wn \\ 0 & 0 \le j < wn\end{cases}$$

基本步骤：

计算 m(I,j)的矩阵，根据跳跃点选取物品。

算法的时间：当$c \le 2^n$ 时，算法需要 O(nc)计算时间

当$c > 2^n$,算法需要$\Omega(n^{2n})$



3、回溯法

策略：选择一个合适的可以搜索的解空间，使用深度优先遍历，使用剪枝算法避免无效搜索，最终求解最优值。

步骤：用 0–1 背包问题子集树构造解空间，其中约束函数的总重量不能超过限制重量，界限函数为 cp+r<=bestp,满足则减去右子树(cp 为当前价值，r 为当前剩余物品价值总和，bestp 为当前最优价值)最后得到 bestp

时间：算法上界时间为 O(n),最坏时候为 $O(2^n)$,算法的时间复杂度为 $O(n2^n)$

4、分支限界法

策略：解空间与剪枝函数与回溯法相同，不同的策略在于分支限界法采用广度优先遍历对解空间进行搜索，并且搜索下一节点的时候可能对下一层级的所有节点进行重要度排序，优先队列式分支限界法。

步骤：解空间与剪枝函数的设计步骤与回溯法相同，而在优先队列中使用各物品　　依照其单位重量价值从大到小排好序，当所有的节点计算出栈以后，就可以得到从根节点到某叶节点的最优解，这个解在结束之前是一直被所找到的比起更好的解替换的。

时间：算法的时间复杂度为 O(n)