**注意：此题目供同学复习课堂内容，和期末考试没有直接关联**

习题2

1. T1和T2是两棵有序树，其中每个结点都有一个标签，考虑树上的三种操作，删除一个子树、插入一个子树和更改一个结点的标签，请设计一个算法，求得从T1变化到T2所需要的最少操作数，要求写出递推方程，程序伪代码并分析时间复杂性。

2. 给定三个字符串A, B和C，设计一个多项式时间动态规划算法，求出它们的最长公共子序列，要求写出递归方程，算法伪代码并分析算法复杂性

3. 考虑字符串变换操作，增加一个字符，删除一个字符以及修改一个字符，设增加字符操作的代价为i, 删除字符操作代价为d, 修改字符的代价为m，给定两个字符串S1和S2，设计一个动态规划算法，求得从S1变换到S2代价最小的变换序列，要求写出递推方程，程序伪代码并分析算法复杂性。

4. 将一根木棒折成若干份，每折一次的代价是当前这段木棒的长度, 总代价是折这根木棒直到满足要求所需要的所有操作的代价。例如，将一根长度为10的木棒折成四段，长度分别为2, 2, 3, 3，如果先折成长度为2和8的两段，再将长度为8的折成长度为2和6的两段，最后将长度为6的折成长度为3的两段，这些操作的代价是10+8+6=24；如果先折成长度为4和6的两段，在分别将长度为4的折成长度为2的两段、长度为6的折成长度为3的两段，则这些操作的代价是10+4+6=20，比上一种方案更好一些。

该问题的输入是木棒的长度L和一些整数c1,…,cn, 要求将木棒折成长度为c1, …, cn的n段且操作代价最小，请设计动态规划算法解决该问题。

5. 满足递归式F(n)=F(n-1)+F(n-2)和初始值F(0)=F(1)=1的数列称为斐波那契数列。考虑如何计算该数列的第n项F(n)。（1）说明根据递归式直接完成计算，将有子问题重复求解；（2）说明该问题具有优化子结构；（3）写出求解F(n)的动态规划算法，并分析算法的时间复杂性。

6. 输入是具有n个数的向量x，输出时输入向量的任何连续子向量的最大和，要求写出递归方程、伪代码并分析时间和空间复杂度。

7. 令I1, …, In是n个区间，其中任一区间Ii=(ai,bi)，假设这些区间按照bi从小到大排序，每一个区间有一个权重vi.找一个互不相交区间的集合，使得这些区间的权重之和最大，例如I1 = (1,2), v1=0.9; I2 = (2,3), v2=0.5; I3 = (1,4), v3=4; I4 = (4,5), v4=2，解是{I3, I4}。

给出解决问题P2的动态规划算法，要求写出递归方程和伪代码，并分析算法时间空间复杂性。

8. 在一个圆形操场的四周摆放着n堆石子，现要将石子有次序地合并成一堆。规定每次只能选择相邻的两堆石子合并成新的一堆，并将新一堆石子数记为该次合并的得分。试设计一个动态规划算法，计算出将n堆石子合并成一堆的最小得分和最大得分，要求列出递归方程，写出算法的伪代码并分析算法的时间空间复杂性。

9. 给定两个字符串s1, s2，其上的操作包括增加一个字符、删除一个字符、修改一个字符和交换两个相邻的字符，其中增加和删除一个字符和交换相邻字符的代价均为1,将字符a修改为字符b的代价记作Ca,b，写出一个动态规划算法求出从s1变化为s2代价最小的变化序列，要求写出递推方程和伪代码并分析时间复杂性。

10. 设有n种不同面值的硬币，面值分别为c1, c2, …., cn分钱, 求用最少个数硬币来找K分钱的策略。要求写出递归方程、伪代码并分析时间和空间复杂度。

11. 给定两个字符串x=x1x2…xn和y=y1y2…ym, 其长度为k的公共子串定义为：x[i,…,i+k-1]=y[j, …, j+k-1]

令kmax为x和y的最长公共子串，设计时间复杂性为O(mn)的算法求kmax。

12. 考虑编辑距离的一种变形，其允许在字符串后无代价地插入无限多个字符，该编辑距离描述为:

ed'(A, B)=min{ed(A, C)|C是B的前缀}, 其中函数ed()是普通的编辑距离函数。根据要求设计算法，要求算法的时间复杂性都是O(|A||B|)

(1) 设计算法，对于给定的字符串A和B，计算ed'(A, B)；

(2) 设计算法，对于给定的字符串A,B和整数k，判定是否B存在某个后缀B’，满足ed’(A, B’)≤k。

13. 我们考虑将数轴上的*n*个点聚成*k*类的问题。

输入：*n*个从小到大的不同实数*x*1, *x*2, …, *xn*表示*n*个不同点，一个参数*k*≤*n*.

任务：将*n*个点划分成*k*个不相交的非空集合*S*1, …., *Sk*满足={*x*1, *x*2, …, *xn*}，*Si*中所有点在*Si*+1中所有点左边，1≤*i*<*k*，也就是说对于任意*x*∈*Si*, *z*∈*Si*+1, *y*<*z*.

目标：最小化,其中cost(*Si*)=(max(*Si*)-min(*Si*))2. max(*Si*)是*Si*中的最小元素，min(*Si*)是*Si*中的最大元素。

例如，如果*Si*={xj}，cost(*Si*)=0，如果*Si*={*xj*, *xj*+1, …, *xj+t*}, *xj*,<*xj*+1< …< *xj+t*，那么cost(*Si*)=(*xj+t*-*xj*)2.

设计时间复杂度为*O*(*n*2*k*)的动态规划算法，找到最优聚类。要求写出伪代码、递归方程并分析算法的时间复杂度。

例如，考虑将4个元素的集合{1,5,8,10}聚为两个类，有三种可能:

1. S1={1}, S2={5,8,10}，总代价是02+52=25
2. S1={1,5}, S2={8,10}，总代价是42+22=20
3. S1={1,5,8}, S2={10}，总代价是72+02=49

所以，算法的解是最优解S1={1,5}, S2={8,10}。

14. 输入一个正整数集合*S*={*x1*, *x2*, …, *xn*}和一个正整数*M*,设计算法判定是否存在*S*的子集合*S*’，使得*S*’中整数之和为*M*。要求写出算法伪代码。