BTH004-最长递增子序列

姓名：赵水 学号：201806150329

1 问题：

对于给定数组，求其最长的递增子序列的长度

2 算法：

采用DP（动态规划）算法。遍历整个数组，并储存（记忆）遍历到每个位置时的**当前**最大递增子序列的长度，不断递归，直至遍历完整个数组，返回储存（记忆）里的最大值

3 代码：

```
import  sys

def longestIncreaseNums(A):
    solutions = []

    dp = [0]
    solutions.append([])
    #插入最小值，便于递归
    A.insert(0,-sys.maxsize-1)
    n = len(A)
    for i in range(1,n):
        index = 0
        for j in range(0,i):
            if A[i] > A[j] and dp[j] > dp[index]:
                index = j
        dp.append(dp[index]+1)
        temp = solutions[index].copy()
        temp.append(A[i])
        solutions.append(temp)

    maxL = max(dp)
    resSolu = []
    for sol in solutions:
        if len(sol) == maxL:
            resSolu.append(sol)

    return resSolu,maxL
```

4 测试

4.1 代码

```
A = [3,5,4,3,6,9,5,7,13,2,15,14]
print(longestIncreaseNums(A.copy()))
```

4.2 输出

```
([[3, 5, 6, 9, 13, 15], [3, 5, 6, 9, 13, 14]], 6)
```