现代密码学作业——第三讲

1、$O(n^2)$ 不是多项式的

2、

（1）

```
def super_increasing_knapsack(A, S):
    def backtrack(curr_set, curr_sum, index):
        if curr_sum == S:
            solutions.append(curr_set)
            return
        if curr_sum > S or index >= len(A):
            return
        for i in range(index, len(A)):
            backtrack(curr_set + [A[i]], curr_sum + A[i], i + 1)

    solutions = []
    backtrack([], 0, 0)
    return solutions
```

（2）O(n*S)

3、

我国密码行业标准 SM4 的密钥长度为 128 位。

现代的个人电脑 CPU 的运算速度大约在几十 GHz 到几百 GHz 之间。为了进行估算，我们可以假设一个中等性能的个人电脑 CPU 每秒可以进行 10^{10} 次操作。

穷尽搜索对于 128 位密钥，可能的密钥总数为 2^{128}。要计算最坏情况下获得密钥所需的时间，我们需要将可能的密钥总数除以每秒可以尝试的密钥数量：

$$所需时间= 2^{128}/10^{10}$$

这个数字是非常巨大的，因此目前通过穷尽搜索来破解 128 位密钥是不现实的。