cs3821 Assignment 3

Jennifer Xu (z5207930)

Question 1

1.1 Time Complexity

The amount of calculations required is

```
n + 2*(n-1) + 2^2*(n-2) + 2^3*(n-3) + ... + 2^n* (1)
```

Therefore, our Time Complexity in Big-O notation is O(2^n).

1.2 Pseudo Code

```
// Store the computed value in a table of size 'n'
table = [ ]

// Fill the table with -1 to indicate the value has not been found yet
for i in range(n): table.append(-1)

function memoized-cut-rod(P, n):
    if n = 0: return 0
    q = -infinity
    for i in range(n):
       if (table[n - i] == -1): table[n - i] = memoized-cut-rod(P, n - 1)
       return max(q, P[i] + table[n - i])
```

1.3 Time Complexity of Memoized Function

The amount of calculations required is

$$n + (n - 1) + (n-2) + \ldots + 3 + 2 + 1$$

Therefore, our Time Complexity in Big-O notation is O(n^2).

1.4 Proof of Correctness

The memoized function we have created has the same results as the original cut-rod function.

The memoized function terminates as we have the same base case where if n equals 0, we return 0.

We call ‘max(q, P[i] + cut-rod(P, n - 1)’ replacing ‘cut-rod(P, n - 1)’ with it’s stored value table[n]. The answer will always be the same since we store the answer away instead of running the function again. Since running the same function over and over again will not result in a different answer since we have a constant P. Therefore, we will achieve the same answer in the end.