James Stallkamp

CS325 Assignment3

1A: see python

1B: see python

1C: The top down approach breaks the problem into smaller pieces. It remembers which cases it has tried to save time and works its way to down to the smallest sub problem. The bottom up approach is the opposite, starting with the smallest sub problem and building up to the final answer.

1D: Both the time complexity and space complexity are O(n^2).

1E: Both the time complexity and space complexity are O(n^2).

1F: The subproblem is the same. T(n) = O(n^2).

2A: DP Pseudocode:

```
int dp[n+1]

dp[0] = 1;

dp[1] = 1; for both of theses cases there is only 1 solution. Base cases

dp[i] = dp[i-1]+dp[i-2] for i between 2 and n.
```

2B: Brute Force Pseudocode:

```
foo()

if n <=1

return 1

else

return foo(n-1)+foo(n-2)
```

2C: For the DP approach the loop runs from 2 to N so the complexity is O(n). For the brute force approach the function makes two recursive calls so the relationship is T(n) = T(n-1)+T(n-2)+1. The one is for the base case call. This comes to O(2^N). The brute force method is dramatically slower.

2D: $T(n) = T(n-1)+T(n-2)+1$