

# Midterm: ENSF 594

Ziad Chemali

Date: 07/08/2020

UCID: 10109966

## Q1) Money Change:

This question is solved using bottom-up dynamic programming.

Finding the minimum number of coins for integer  $n$ , first create an array of size equal to  $n+1$ .

At each index of the array, the minimum number of coins is evaluated, and stored in the array.

After all indexes have been evaluated, the last index will contain the minimum number of coins for integer  $n$ .

```
Enter an integer between 1 and 10^3
34
Minimum number of coins is : 9
```

## Q2) Primitive Calculator:

This question is also solved using bottom-up dynamic programming.

Finding the minimum number of operations for integer  $n$  starting from 1, first create an array of size equal to  $n+1$ .

At each index of the array, the minimum number of operations is evaluated, and stored in the array.

After all indexes have been evaluated, the last index will contain the minimum number of operations for integer  $n$ .

Also, a map is initiated to store the type of operation at each index. This map is used to print the intermediate numbers between 1 and  $n$ .

```
Enter integer between 1 and 10^6
96234
The minimum number of operations is: 14
[1, 3, 9, 10, 11, 33, 99, 297, 891, 2673, 8019, 16038, 16039, 48117, 96234]
```

## Q3) Quick Sort:

Quick sort is modified so that it separates the array to elements greater than pivot, and elements smaller than pivot, and what is left is just elements equal to pivot.

This is done by using a forward and backward pass over the array.

The forward pass will swap any element smaller than pivot

The backward pass will swap any element greater than pivot

Both forward/backward passes are invoked until the entire array is sorted.

```
Enter Array size
6
Enter element: 1
2
Enter element: 2
2
Enter element: 3
3
Enter element: 4
3
Enter element: 5
7
Enter element: 6
1
Sorted array:
1 2 2 3 3 7
```

#### Q4) Signature:

This question is solved by first arranging all the segments based on the end points. Then compare the start point of each segment to pivot (Note: pivot is initialized to min start point -1), if the start point is greater than pivot then include the segment's end point as a common point. Keep doing the same thing until all segments are covered.

```
How many segments
4
Enter first pair of element1
4
Enter second pair of element1
7
Enter first pair of element2
1
Enter second pair of element2
3
Enter first pair of element3
2
Enter second pair of element3
5
Enter first pair of element4
5
Enter second pair of element4
6
The common points between segments are:
[3, 6]
```