DESIGN AND ANALYSIS OF ALGORITHMS – CSCI 612

NAME: AADITHYA GOPALAKRISHNA BHARADWAJ

ZID: Z1862641

ASSIGNEMENT NUMBER, SEMESTER: Assignment 4, SPRING 2019

==========

1.) For the given array {3, 4, -5, 2, -3, 3, -1, 7, -2, 1}, find the maximum subarray using the dynamic programming algorithm introduced in class. Show steps.

Answer.)

Given Array {3, 4, -5, 2, -3, 3, -1, 7, -2, 1} => Length is 10

Let MF = Max_So_Far = 0

Let ME = Max_Ending_Here = 0

For i = 0, a[0] = 3

ME = ME + 3 => ME = 3

Since ME is greater than MF, therefore MF = ME => MF = 3

For i = 1, a[1] = 4

ME = ME + 4 => ME = 7

Since ME is greater than MF, therefore MF = ME => MF = 7

For i = 2, a[2] = -5

ME = ME + (-5) => ME = 2

For i = 3, a[3] = 2

ME = ME + 2 => ME = 4

For i =4, a[4] = -3

ME = ME + (-3) => ME = 1

For i =5, a[5] = 3

ME = ME + 3 => ME = 4

For i = 6, a[6] = -1

ME = ME + (-1) => ME = 3

For i =7, a[7] = 7

ME = ME + 7 => ME = 10

Since ME > MF => MF = ME => MF = 10

For i =8, a[8] = -2

ME = ME + (-2) => ME = 8

For i = 9, a[9] = 1

ME = ME +1 => ME =9

Hence from the above we find that MF is highest at i = 7, where MF is 10

Therefore, the maximum subarray is: {3, 4, -5, 2, -3, 3, -1, 7} with sum = 10

==========

2.) Longest palindrome subsequence

A palindrome is a nonempty string over some alphabet that reads the same forward and backward. Examples of palindromes are all strings of length 1, "civic", "racecar", and "aibohphobia" (fear of palindromes). Design a dynamic programming algorithm to find the longest palindrome that is a subsequence of a given input string. For example, given the input "character", your algorithm should return "carac".

Answer.)

```
//dynamic programming
//longest common subsequence method

string lcs(string X, string Y)
{
	int m = X.length();
	int n = Y.length();

	int L[m+1][n+1];

	/* Following steps build L[m+1][n+1] in bottom up fashion. Note that L[i][j] contains
	length of LCS of X[0..i-1] and Y[0..j-1] */

	for (int i=0; i<=m; i++)
	{
		for (int j=0; j<=n; j++)
		{
			if (i == 0 || j == 0)
				L[i][j] = 0;
			else if (X[i-1] == Y[j-1])
				L[i][j] = L[i-1][j-1] + 1;
			else
				L[i][j] = max(L[i-1][j], L[i][j-1]);
		}
	}
```

```
	// Following code is used to print LCS

	int index = L[m][n];

	// Create a string length index+1 and fill it with \0

	string lcs(index+1, '\0');

	// Start from the right-most-bottom-mostcorner and one by one store characters in lcs[]

	int i = m, j = n;
	while (i > 0 && j > 0)
	{
		// If current character in X[] and are same, then current character is part of LCS

		if (X[i-1] == Y[j-1])
		{
			// Put current character in result

			lcs[index-1] = X[i-1];
			i--;
			j--;
			// reduce values of i, j and index
			index--;
		}
		// If not same, then find the larger of two and go in the direction of larger value
		else if (L[i-1][j] > L[i][j-1])
			i--;
		else
			j--;
	}
return lcs;
}

// Returns longest palindromic subsequence of str

string longestPalindromicSubsequence(string &str)
{
	// Find reverse of str
	string rev = str;
	reverse(rev.begin(), rev.end());
/* Return LCS of str and its reverse here LCS of str and its reverse is the longest   palindromic subsequence */
	return lcs(str, rev);
}
```

3.) Coin changing:

Consider the problem of making change for n cents using the fewest number of coins given any set of k different coin denominations. For example, coins in America are in denominations of 1, 5, 10, and 25 cents. However, this question does not have any specific denomination. It only assumes that each coin's value is an integer and one of the coins is a penny.

Answer.)

1.) Declare Vaiables : Quarters, Dimes and Cents

2.) /* calculate number of quarters*/

```
if(n/25>=1)
        quarters = n/25;
        n = n-(quarters*25);
```

3.) /* calculate number of dimes*/

```
if(n/10>=1)
        dimes = n/10;
        n = n-(dimes*10);
```

4.) /* calculate number of nickels*/

```
if(n/5>=1)
        nickels = n/5;
        n = n-(nickels*5);
```

5.) /* calculate number of cents*/

```
if(n>=1)
        cents = n;
```

6.) Print "1 cent coin(s)----->pennies"