Berkcan
Altungöz
20170808014

Question 1

a) $$C(n) = \begin{cases} 0 & \text{if } n = 0 \\ \min(1 + C(n - w_i)) & \text{if } n > 0 \end{cases}$$

$C(n)$ = min no of weights $w_1, w_2, w_3 \dots w_k$
Let w_i be the first weight where $w \leq n$

c)
```
int minWeight(weight[], length, n)
    if n == 0 {
        return 0
    }
    else {
    result = INT_MAX
    for (i = 0; i < length; i++) {
        sub;
        if(weight[i] >= n {
        sub = 1 + minWeight(weight[], length, n - weight[i])
        result = min(result, sub)
        }
    }
    return result
}
```

b) The recursive call
minWeight(weight[], length, n) = min(result, 1 + minWeight(weight[], length n - weight[i]) will generate a recursion tree that will create overlapping subproblems.

Berkcan Altungöz
20170808016

e)
```
int MinWeight (weight[], dp[], length, n) {
    if (dp[n] != -1) {
        return dp[n]
    }
    if (n == 0) {
        return 0
    }

    result = INT_MAX
    for (i=0; i < length; i++) {
        if (coins[i] <= n) {
            sub = 1 + min weight(weight[], dp, n-coins[i])
            if (result > sub) {
                result = min (result, sub)
            }
        }
    }
    dp[n] = result
    return dp[n]
}
```

d) For values 1, 3, 4. For n=6 the optimal would be 3+3 so C(n) would be 2. But greedy algorithm will give 4+1+1 so C(n) would be 3. The greedy leads to a non optimal solution.

f) Running time is O(length * n)

Berkcan Altungöz
20170808014

Question 2

b) Let $\Phi_i = i \bmod n$

when $i \bmod n = 0$,

$a_i = n + 0 - (n-1) = 1$

when $i \bmod n \neq 0$,

$a_i = 1 + (i \bmod n) - ((i-1)) \bmod n = 2$

Question 3

```
lcs ( A[], B[], i, j) {
 i = 0 , j = 0
 m = len ( A[] )
 n = len ( B[] )
 if i >= m  or j >= n {
     return 0
     }
 else if ( A[i] = B[j] {
    return 1 + lcs(A[], B[], i+1, j+1 )
 }
return max ( lcs ( A[], B[], i, j+1) , lcs (A[], B[], i+1, j ))
}
```