Assignment-5

Ans-1) Here a dp is used to solve the following. Where each element of dp stores a pair where the first is the minimum number of coins required to attain the given change and the second is used to store which coin was last used to attain the given change.
Initially except 0 all other changes are set to INF=10^9, indicating that making the given change is not possible. Then we forward iterate the dp from 0 to X and mark the element setting the minimum coins to make [current value + a[i]] required to 1+dp[current value].first. And thereby storing the last element (a[i]) used the make the given sum.
N – The number of denominations of coins
X – The required change of coin

Table formed while traversing:-

Sorted a[n]={1,2,5,6,8}

X\N	0	a[1]=1	a[2]=2	a[3]=5	a[4]=6	a[5]=8
0	dp[0]=[0,0]	dp[1]=[1,1]	dp[2]=[1,2]	dp[5]=[1,5]	dp[6]=[1,6]	dp[8]=[1,8]
1	dp[1]=[1,1]	skip	dp[3]=[2,2]	skip	dp[7]=[2,6]	dp[9]=[2,8]
2	dp[2]=[1,2]	skip	dp[4]=[2,2]	skip	Skip	dp[10]=[2,8]
3	dp[3]=[2,2]	skip	skip	skip	skip	dp[11]=[2,6]
4	dp[4]=[2,2]	skip	skip	skip	skip	Break
5	dp[5]=[1,5]	skip	skip	skip	dp[11]=[2,6]	Break
6	dp[6]=[1,6]	skip	skip	skip	Break	-
7	dp[7]=[2,6]	skip	skip	Break	-	-
8	dp[8]=[1,8]	skip	skip	Break	-	-
9	dp[9]=[2,8]	skip	skip	Break	-	-
10	dp[10]=[2,8]	skip	Break	-	-	-
11	dp[11]=[2,6]	-	-	-	-	-

Time Complexity – O(N*X)
Space Complexity – O(N+X)

Ans-2) To perform in-place merge-sort the dividing part remains the same, but merging two sorted array is different. Here we have to merge two sorted array in O(1) Space- Complexity.

Two Sorted array can be merged in O(1) space using the concept of shell sort but here instead of O(N) merging time it takes O(N logN) time.

In thr merge() function, we initially set a gap of ceil(len,2) and half the gap each time. In each gap we have 2 pointers in which the first one points at the start of array and the second one points to the location at gap+first_pointer and the elements at the given positions are compared and if the element at second pointer is greater then it is swapped with the first_pointer element.

The gap takes $O(\log_2 N)$ time to come to 1 and we break the loop in that case.

Hence the,

Overall time complexity – $\mathbf{O(N \log^2 N)}$

Space Complexity – **O(1)**