Question 1 solution

In greedy Algorithms, we make whatever choice seems best at pre moment and then solve me sub problems arising after the choice is made.

Question 2 solution

Recurssive Algo for activity-selection problem

RAS (S, f, k, n) {

     m = k + 1

        while m ≤ n and S[m] < f[k]

            m = m + 1

       if m ≤ n

          return { m } ∪ RAS (S, f, m, n)

       else

          return 0

}

Question 3 Solution

Any change we make to the greedy algo, Simply change the way of reading the target value. If does not account for the minimum coins used. To put better in a better way a strict move does not exented for this problem. A higher denenation coin may yield target

quickly   but   it   is   not   safe   move

Question 4   Solution.

Given   a:1 , b:1 , c:2 , d:3, e:5, f:8
        g:13 . h:21 , Since   move   are   8
letters   in   alphabet, the   intial   queue size
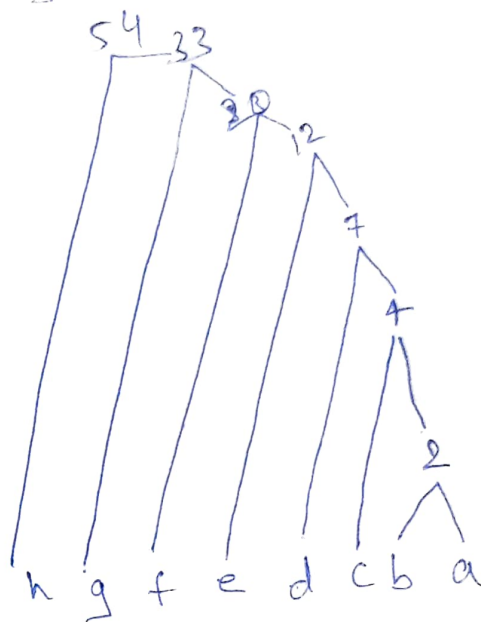is   n = 8   and 7   merge   steps   are required.

h : 0

g : 1 0

f : 1   1 0

e : 1   1   1   0

d : 1   1   1   1   0

c : 1   1   1   1   1   0

b : 1   1   1   1   1   1   0

a : 1   1   1   1   1   1   1

54
33
20
12
7
4
2

h g f e d c b a

Question 5   Solution.

fractional knapsack Algo :-

Step1 : for each   item , compute   its   value/
       weight ratio.

step2 : Arrange   all   the   items   in decreasing
       order   of   their   value/wt ratio.

Step 3 : Start putting item   into   the knapsack
       log from   the   item   with   the
       heighest   ratio. put   as   array items
       as   we   can into   the knapsack.
       The complexity : $O(\lg n)$ & due to sorting