Name: Prathamesh Zonkar
BNumber: B00811727.

Home work 4.

~~Q2a~~)
Q1)

i) Suppose if we have activities with start time as follows

~~{(2, 11) (12, 20) (4,19)}~~ (

{(1, 2) (2, 7) (3, 10) (4, 19) (11, 15) (16, 17)}

Here if we pick the activity (4,19) as it is the most compatible and ~~does not overlap with any~~ has the least conflict we are ignoring the ~~other~~ optimal solutions
of (1,2) (3,10) (11, 15) (16, 17).

ii) Suppose we have the activities a follow
{(2, 15) (3,4) (5, 6) (7, 8) (9, 12) (12, 16)}

If we choose the activity with the ~~most~~ earliest start time i.e (2,15) we will not get the optimal solution with the greedy approach.

Q2)

a) Describe a greedy algorithm to make change consisting of quarters, dimes, nickels & pennies. Prove that your algorithm yields an optimal solution.

The pseudo code for such an algo will be as follows:

for (int i = size of denomination array to i = 0.

do while (denom [i] ≤ amount)

change [j] ← denom [i].

j++;

amount = amount - denom [i];

count ++.

return count.

Here we can see that as we are using the largest denomination first to make change of the amount, by using the greedy strategy we will subtract the larger denomination to reduce the amount by great value hence we will eventually be using less number of coins ie instead of using 5 pennies we will use one nickel and so on. Hence this algorithm will yield an optimal solution.

Q2 b) Suppose that the available coins are in the denomination that are powers of c, ie the denomination are $c^0, c^1 \ldots c^k$ for some integer $c > 1$ & $k \geq 1$. Show that the gredy algorithm always yields an optimal solution.

Ans - We will use the ~~algo~~ pseudo code similar to the previous algo.

- As we are having the denomination in an increasing order of c^k our denomination will be similar to previous denomination just in the power of c.

- Hence by using the previous algorithm too we will get an optimal solution as we will use up the greater ~~denom~~ denominations first to calculate the ~~cop~~ change which will result in lower coin count.

Scanned with CamScanner

Q2)
i) Give a set of coin denomination for which the greedy algorithm doesnot yield and optimal sollution. your set should include a penny so that there ~~&~~ is a sollution for every value of n.

Let us consider the denominations as {1, 4, 5}
If we want to calculate the change for amount 12.

We can see that by our algorithm we get the denomination as

$$2 \times 5 + 2 \times 1 = 12.$$

when we use ~~&~~ two 5 & ~~2~~ two 1 denomination ~~where~~ as which totals to 4 coins

Where as there is an optimal solution which is ~~3~~ three 4 denomination.

Hence for this set it won't give an optimal solution.