Evan Zhang | z5383657

Question 2

2.1

Number of customers n = 3.

For n = 1, $s_1 = 1$, $e_1 = 100$, $a_1 = 1$, $b_1 = 1$ and $c_1 = 1$.

For n = 2, $s_2 = 2$, $e_2 = 100$, $a_2 = 1$, $b_2 = 1$ and $c_2 = 1$.

For n = 3, $s_3 = 3$, $e_3 = 100$, $a_3 = 1$, $b_3 = 1$ and $c_3 = 100000$.

Running the greedy algorithm, we process each customer in order of arrival.

For customer 1, $a_1 = b_1$ so we assign that customer to Alice.

For customer 2, $a_2 = b_2$, so we assign that customer to Bob.

For customer 3, there is no mechanic available, as both Alice and Bob are busy until they finish their current job, which ends at $e_1 = e_2 = 100$. So, customer 3 will not be served.

Customer 1 and customer 2 will make 1 dollar each having been served by Alice and Bob respectively. Customer 3 will lose the store 100000 dollars having not been served at all. Thus, the greedy algorithm will lose the store 99998 dollars total.

A higher net figure can be made by cancelling the job Alice/Bob is on and serving the third customer. When doing this, the total earned will remain the same (2 dollars) while the total lost will reduce to 1 dollar. Thus, the net earnings will be 1 dollar, which is higher than the greedy algorithm's net earnings of negative 99998 dollars.

2.2

<u>Subproblem:</u>

For $0 \le i,j \le n$, let P(i ,j) be the problem of determining opt(i ,j), the maximum total earnings that can be achieved using up to i customers and using only the first j customers.

<u>Recurrence:</u>

opt(i,j) = max(opt(i, j - 1) - c_i, opt(i, j - 1) + a_i, opt(i, j - 1) + b_i, opt(i - 1, j - 1) + a_i - (a_k + c_k), opt(i - 1, j - 1) + b_i – (b_k + c_k)).

There are five options when processing a new opt(i):

1. Nobody is available and you don't serve the new customer, losing c_i dollars.
2. Alice is available and serves the new customer, making a_i dollars.
3. Bob is available and serves the new customer, making b_i dollars.
4. Alice doesn't serve her current customer and serves the new customer, losing (a_k + c_k) dollars and making a_i dollars.
5. Bob doesn't serve his current customer and serves the new customer, losing (b_k + c_k) dollars and making b_i dollars.

<u>Base case:</u>

opt(0,0) = 0, Alice and Bob don't have any current customers.

Order of computation:

Ascending order of j then i. This guarantees subproblems are completed when we reach P(i,j).

Final answer:

opt(n,n).

Time Complexity:

$O(n^2)$ since for each i, order O(1) steps are carried out j times.