Question 2

2.1

Number of customers n = 3.

For n = 1, s1 = 1, e1 = 100, a1 = 1, b1 = 1 and c1 = 1.

For n = 2, s2 = 2, e2 = 100, a2 = 1, b2 = 1 and c2 = 1.

For n = 3, s3 = 3, e3 = 100, a3 = 1, b3 = 1 and c3 = 100000.

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

For customer 1, a1 = b1 so we assign that customer to Alice.

For customer 2, a2 = b2, 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 e1 = e2 = 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

Subproblem:

For 0 ≤ i,j ≤ 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.

Recurrence:

opt(i,j) = max(opt(i, j - 1) - ci, opt(i, j - 1) + ai, opt(i, j - 1) + bi, opt(i - 1, j - 1) + ai - (ak + ck), opt(i - 1, j - 1) + bi – (bk + ck)).

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

1. Nobody is available and you don’t serve the new customer, losing ci dollars.
2. Alice is available and serves the new customer, making ai dollars.
3. Bob is available and serves the new customer, making bi dollars.
4. Alice doesn’t serve her current customer and serves the new customer, losing (ak + ck) dollars and making ai dollars.
5. Bob doesn’t serve his current customer and serves the new customer, losing (bk + ck) dollars and making bi dollars.

Base case:

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(n2) since for each i, order O(1) steps are carried out j times.