2.a.i) Given that you have the 4 options from the table 1, the “next available location” algorithm would place the minimarts in the configuration depicted by table 2. The total for this configuration is 34, but this is not an optimal solution since we can achieve a total of 38 with the configuration depicted by table 3.

|  |  |
| --- | --- |
| x | r |
| 1 | 1 |
| 7 | 10 |
| 8 | 14 |
| 13 | 23 |

*Table 1*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| x(1) = 1 | x(7) = 10 | x(14) = 23 |  | Total equal = 34 |

*Table 2*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| x(1) = 1 | x(8) = 14 | x(14) = 23 |  | Total equal = 38 |

*Table 3*

2.a.ii) Given that you have the 4 options from table 4, the “most profitable first” algorithm would place the minimarts in the configuration depicted by table 5. The total for this configuration is 24, but this is not an optimal solution since we can achieve a total of 37 with the configuration depicted by table 6

|  |  |
| --- | --- |
| x | r |
| 1 | 1 |
| 7 | 15 |
| 12 | 23 |
| 13 | 21 |

*Table 4*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| x(12) = 23 | x(1) = 1 |  |  | Total = 24 |

*Table 5*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| x(1) = 1 | x(7) = 15 | x(13) = 21 |  | Total = 37 |

*Table 6*

2.b.i) The subproblems will be denoted by OPT(j) and OPT(j) represents the optimal profit for minimart 1,2… j.

2.b.ii)

* add a new list of finishing time call such that
* sort all the data such that
* let p(j) be the largest index k < j that is minimart k is compatible with j
* Given all this, the recurrence relation is
* Proof:
  + Assume that is an optimal price and , our proof relies on 2 properties:
    - When looking at minimart j, the only 2 options are whether to include or not include minimart j
      * If included, OPT(j) would include the price of minimart j plus the optimal price of the minimart that minimart j is most compatible with, which is
      * If not included, would be equal to , which is and from our original statement, and we know that is optimal already
    - The max function checks and ensure that has better price or the same as
  + Conclusion: due to the 2 properties above, we can safely say that if the algorithm decides to add minimart j then it will improve on

2.b.iii & iv)



* The run time for the MinimartOPT function is O(n), but the sorting and ComputeP algorithms are O(nlogn); therefore, the runtime of the program is O(nlogn)
* Since the MinimartPrint function can potentially call itself n times, the runtime of it is O(n)
  + Note: This runtime does not take into account the run time of the sorting or ComputeP algorithms