CS3230 PA3

Yadunand Prem, A0253252M

Q1

Optimal Substructure

Let j be the number of houses you're visiting and S_j be the smallest sum of anger that is attainable from j houses, A_j be the jth smallest anger that they can attain Then $S_j = A_j + S_{j-1}$

Optimal solution of j houses is the sum of the optimal solution of $j - 1$ houses, and the anger of the house with jth smallest anger.

Proof of Correctness

1. Let S_j be the minimum anger attainable through visitng j houses (Optimal)
2. Suppose that the S_{j-1} solution is not optimal.
3. Then there exists a min anger x of $j - 1$ houses where $x < S_{j-1}$
4. Combining $x + A_j$ would then give a solution that is less than S_j contradicting our initial clause.
5. Thus, this optimal substructure exists

Greedy Choice

Greedy Choice: Let x be the house with the minimum anger. Then there exists an optimal solution containing x, called S.

Proof of Greedy

1. Suppose not, that $i \notin S$
2. Then, we can replace any x in S with i and get a sum with a lower anger (Cannot be same anger as i is distinct). This contradicts the initial statement of S being the optimal solution.

Last Step

1. To get the house with jth smallest anger, we can sort the houses by its anger values. This can be done in $O(n \log(n)$ timing. Without this, it would take $O(k)$ time to get the k-th smallest anger each time.
2. We can then build a prefix-sum on the anger values in $O(n)$ time. Now by accessing the $i - 1$th element of the array (as its 0 indexed), we can immediately retrieve the min anger attainable in i houses in $O(1)$ timing.

Total Time Complexity: $O(n \log(n) + n + 1) = O(n \log(n))$

Q2

Optimal Substructure

Let $S(m)$ be the number of ways that frank can arrange the menu items in m minutes. Then $S(m) = \sum(S(m - T_i))$, where T_i is the time taken for the ith item. To note: repetition is allowed and $S(k) = 0, \forall k < 0)$.

Overlapping Subproblems

There are overlapping subproblems when there are 2 courses with the same time. For example, if there are 2 courses a and b, each with time $T_a = T_b = 1$, then $S(m - T_a) = S(m - T_b)$. Thus, there are overlapping subproblems.

If all subproblems are recomputed, the time complexity would be $O(n^m)$ in the worst case, where n is the number of minutes and m is the number of courses. This has an exponential time complexity.

If the overlapping subproblems are avoided, then this has an complexity of $O(nm)$. The DP problem has 1 parameter m, and in each subproblem, it calculates the sum of n integers, which runs in $O(n)$ time. Thus, $O(nm)$. This has a pseudopolynomial timing