Kick Start 2022 - Round E

Analysis: Pizza Delivery

Test Set 1

Since Ada does not have to deliver any pizzas in this test set, our task is simply to find the most profitable path of $\mathbf{M}$ steps starting from the location $(\mathbf{A_r}, \mathbf{A_c})$. It is important to note that the functions $F_i(c)$ are increasing given that $\mathbf{K_i} \geq 1$. In other words, to maximize the outcome of the functions, we should try maximizing the current number of coins c. Therefore, to find the profit of an optimal path of length $\mathbf{M}$ from $(\mathbf{A_r}, \mathbf{A_c})$ to a specific location (i, j), we must find optimal paths of length $\mathbf{M} - 1$ to all of its neighbouring locations first, apply the respective toll functions F, and take the maximum outcome. And whenever we can reduce a problem to the same but smaller sub-problems, dynamic programming can help most of the time.

Formally, let $DP_{i,j,t}$ be the maximum profit of a path of length t from $(\mathbf{A_r}, \mathbf{A_c})$ to the location (i, j) and let $L_{i,j}$ be the neighbouring locations of (i, j) ($|L_{i,j}| \leq 4$). Then

$$DP_{i,j,0} = \begin{cases} 0 & \text{if } (i,j) = (\mathbf{A_r}, \mathbf{A_c}) \\ -\infty & \text{otherwise} \end{cases}$$

and

$$DP_{i,j,t+1} = \max(DP_{i,j,t}, \max_{(i',j') \in L_{i,j}} DP_{i',j',t} \, \mathbf{OP_*K_*}).$$

For brevity, $\mathbf{OP_*K_*}$ refers to the appropriate function F when going from location (i', j') to location (i, j). Note that Ada can choose to not move at all, hence the term $DP_{i,j,t}$ in the recurrence above.

The final answer is $\max_{1 \leq i,j \leq \mathbf{N}} DP_{i,j,\mathbf{M}}$. The time complexity for computing the entire DP table is $O(\mathbf{N}^2\mathbf{M})$.

Test Set 2

In the case Ada actually has to deliver some pizzas, the problem can be solved in much the same way, except that we need to keep track of the pizzas that have been delivered already. So instead of asking, what is the best way to reach the location (i, j) in t steps, we are wondering, what is the best way to deliver a subset $S \subseteq \{1, 2, \ldots, \mathbf{P}\}$ of pizzas in t steps and end up in the location (i, j). All we have to do is extend our DP table by another dimension, namely, all the possible subsets S of pizzas.

Formally, let $DP_{i,j,S,t}$ be the maximum profit of delivering the subset of pizzas S in t steps and ending up at the location (i, j). Then

$$DP_{i,j,S,0} = \begin{cases} 0 & \text{if } (i,j) = (\mathbf{A_r}, \mathbf{A_c}) \text{ and } S = \emptyset \\ -\infty & \text{otherwise} \end{cases}.$$

As for the recurrence relation, we must account for Ada's choice to deliver pizzas after each step. If none of the customers in S lives at the location (i, j), then delivering a pizza after the next step is not an option, and our recurrence relation remains the same:

$$DP_{i,j,S,t+1} = \max(DP_{i,j,S,t}, \max_{(i',j') \in L_{i,j}} DP_{i',j',S,t} \, \mathbf{OP_*K_*}).$$

However, if, say, customer x lives at the location (i, j) and $x \in S$, then Ada can choose to deliver their pizza after making the next step, therefore, our recurrence relation becomes

$$DP_{i,j,S,t+1} = \max(DP_{i,j,S,t}, DP_{i,j,S-\{x\},t} + \mathbf{C_x}, \max_{(i',j') \in L_{i,j}} DP_{i',j',S,t} \, \mathbf{OP_*K_*}, \max_{(i',j') \in L_{i,j}} DP_{i',j',S-\{x\},t} \, \mathbf{OP_*K_*} + \mathbf{C_x}).$$

The final answer is $\max_{1 \leq i,j \leq \mathbf{N}} DP_{i,j,\{1,2,\ldots,\mathbf{P}\},\mathbf{M}}$ and the time complexity of the algorithm is now $O(\mathbf{N}^2 2^{\mathbf{P}} \mathbf{M})$.