PRINCETON COS CON

Problem 3: Swiping Swag

By Henry Tang

Other than problem 0, this problem appeared to be the easiest and was successfully solved by 14 teams.

We can begin by observing that if we have an efficient way to calculate the maximum number of tables that Twelve can hit for a fixed arm range, we can just binary search in the range $[0, n]$ to find the minimum arm length that accomplishes the task. This is because having longer arms will always guarantee at least the same amount of swag (under optimal choice of teleportation) as having shorter arms. (Why? Just take the same optimal path!)

Let us fix k and let $dp[i]$ represent the maximum number of tables she has taken swag from after reaching table i. We know that $dp[1] = \min(k + 1, n)$. Since it's always optimal to take the maximum amount of swag possible whenever Twelve is at a table, we have the following recurrence for $dp[j]$, for each table j connected to table i via a teleportation path:

- If there is no path to i from 1, then $dp[j] = 0$.
- Otherwise, if there is a teleportation pathway from i to j, then let a be how much of her arm extends past table n if we are currently at table j (that is, $a = \max\{j + k - n, 0\}$). We discern two cases:
  - If $j - i > 2k + 1$, then $dp[j] = \max(dp[j], dp[i] + 2k + 1)$. This is the case where there does not exist a table that Twelve's arms can reach from table i and from table j.
  - Otherwise, $dp]j] = \max(dp[j], dp[i] + j - i - a)$. This is the case where there may be tables that Twelve's arms can reach from table i and table j. In this case, we just have to be careful not to double-count tables that we have already counted when calculating $dp[i]$.

Carrying out the above procedure for every $1 \leq i \leq n$, we can be assured that $dp[i]$ contains the maximum amount of tables Twelve can hit once she reaches table i. At the end, we just need to check whether the maximum of dp is greater that t.

As we said before, all we need to do now is a binary search (with the above as a subroutine) on $[0, n]$ to find the minimum k that allows her to swipe swag from t tables.

Time Complexity: $\mathcal{O}((n + t) \log n)$, as the dynamic programming step takes time linear in n and t, and we have to do up to $\log n$ iterations in the binary search.

PRINCETON COS CON

Plaudits:

- Congratulations to Antonio Molina (grad) for being the first to solve this problem in a *very* impressive 14 minutes!
- Congratulations to Devon Ulrich '23 and Andrew Chen '23 for being the first UG team to solve this problem at an impressive 41 minutes! They won the fastest UG submission prize for their solution to this problem.