1. Explain the strategy of the approximation.

I used 2 different heuristics in combination, one after the other. Nearest Neighbor being used first to establish the initial path, then using Two-opt on that path we just created using Nearest Neighbor, to then create a more optimal path with Two-opt. We avoid slicing as much as possible compared to the exact solution also but slicing is O(k) which can be very bad runtime with big inputs.

2. Discuss the analytical runtime analysis of the approximation.

$O(n)$ (build adjacency dictionary)
+ $O(n^2)$ (Nearest Neighbor)
+ $O(n^2)$ (Two-Opt pair checks)
+ $O(n^3)$ (Two-opt worst case switch)
+ $O(n)$ (Calculate tour length)
= $O(n^3)$ worst case runtime

3. Results of approximation running "reasonably well" on tests in gradescope.

tourcost1=277029.5438   tourcost2=2178