Approximation Notes

Strategy:
The approximation solution uses a combination of a few strategies such as greedy, anytime, restarts, and 2-opt to produce an acceptable answer.

It first starts with greedy, where it goes through the vertices and picks the nearest neighbor (with random tie breaking) for each one. After that, it uses 2-opt to optimize the cycle using edge swaps until no improvements are found. These both are under the anytime time budget with restarts.

Analytical runtime analysis:
The runtime is O(kn^2), where k is the number of restarts that the anytime approximation attempts.
Greedy runs in O(n^2) and 2-Opt runs in O(n^2) with k restarts.

Program performance on Gradescope:
Tour 1: 241710.2954
Tour 2: 2085

run_test_cases.sh test cases:
N = 100: 8.0774
N = 500: 27.6677
N = 1000: 5.9793
Nonoptimal: 280.0000