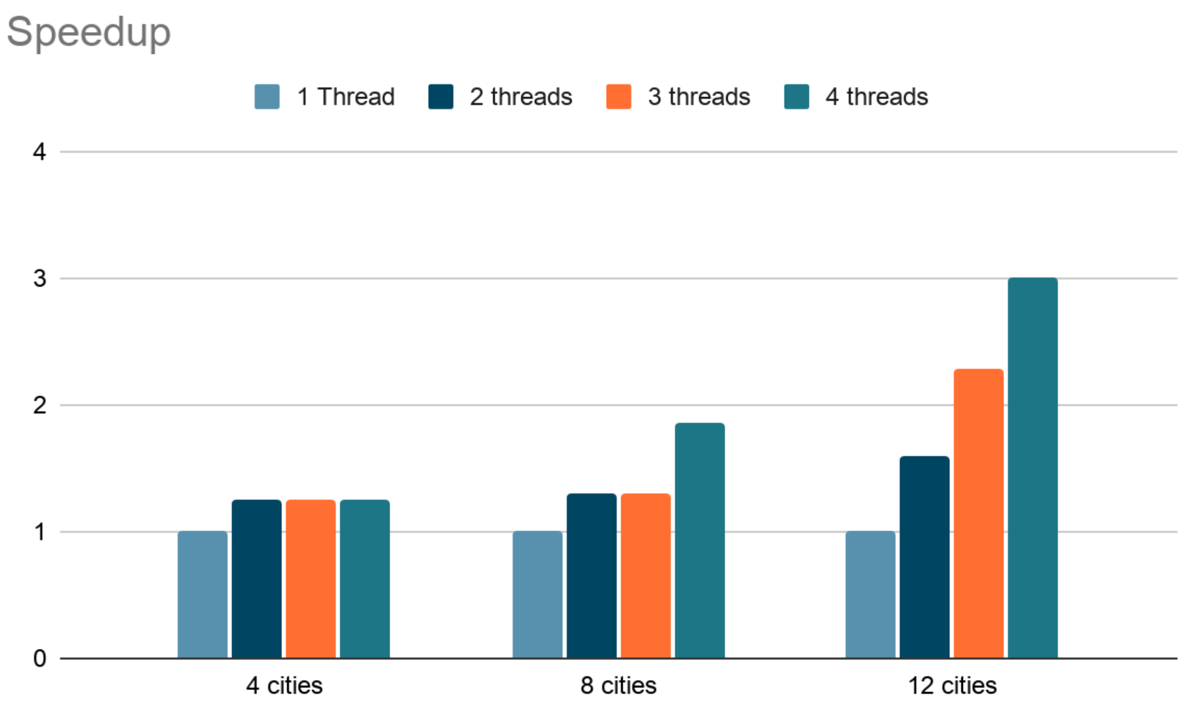
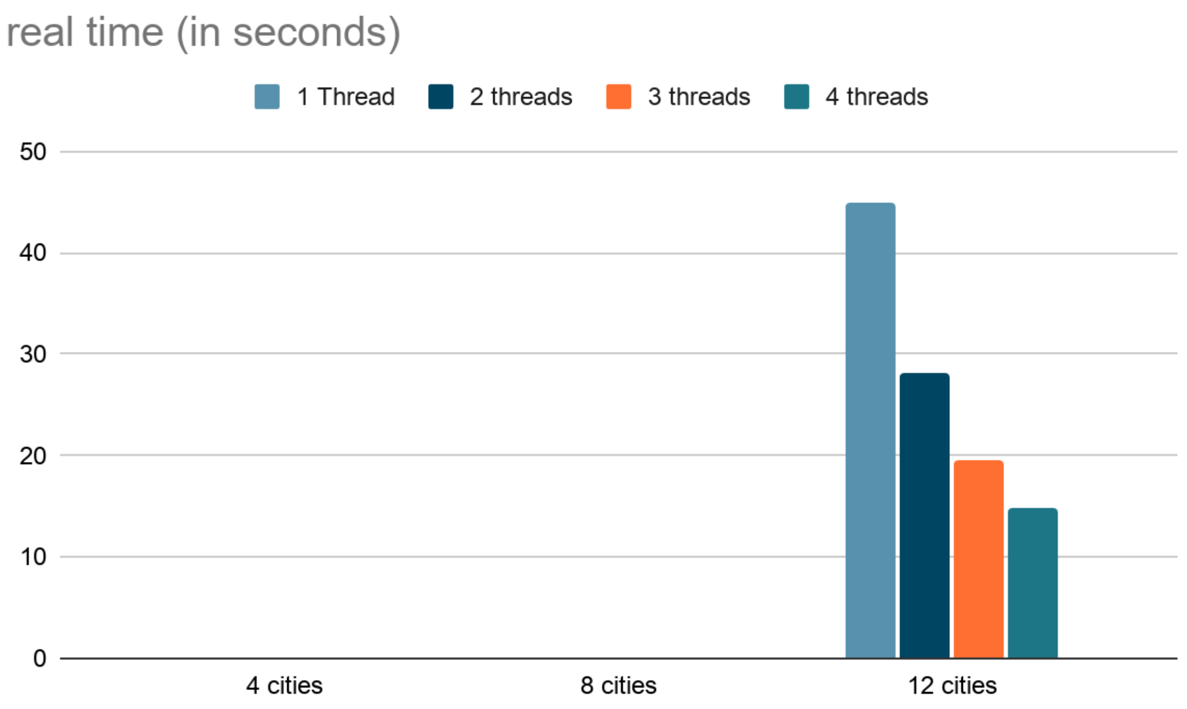
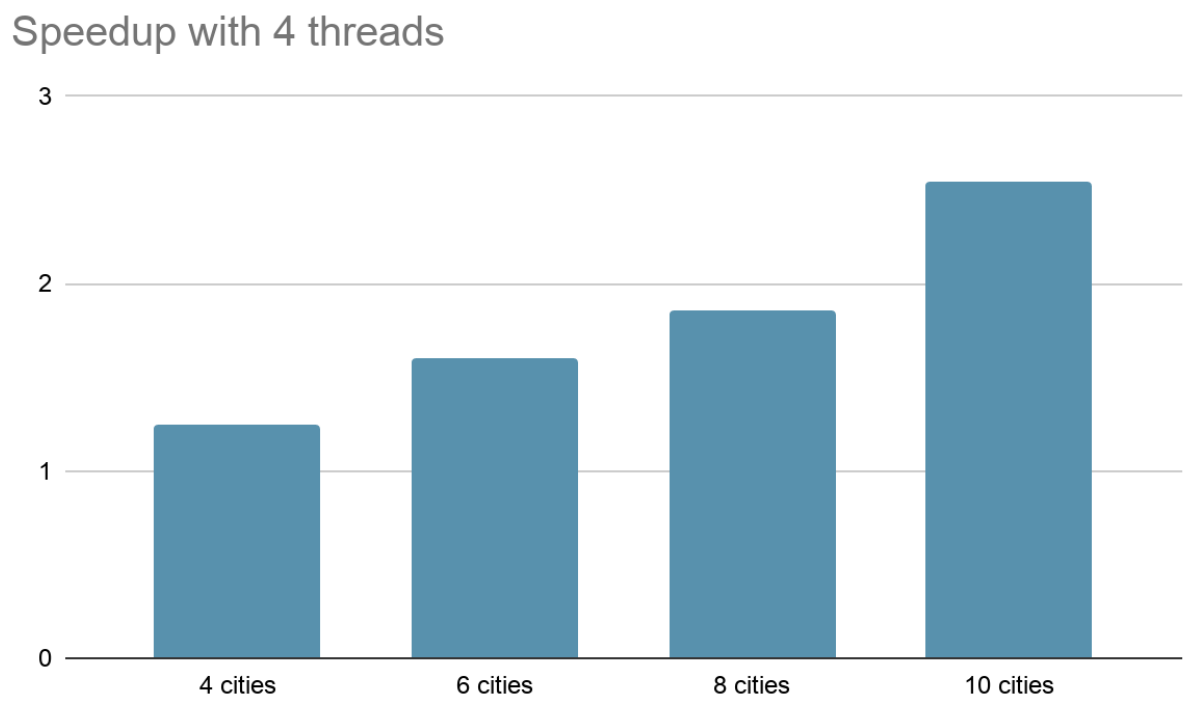
Graph 1: A graph that shows speedup over single-thread version for a problem of size 4 cities, another graph of problem of size 8 cities, and a third graph for 12 cities. For each graph, the x-axis the number of threads (1, 2, 3, and 4) and y-axis is the time generated by time command (real, not user or system).





Graph 2: Fix the number of threads to four. The y-axis is the speedup relative to one thread. The x-axis is the number of cities. In an increment of 2, starting with 4, show the speedup for 4, 6, 8, 10, … x. Where x is the smallest number of cities where four threads show a speedup > 2. If x turns out to be 6 or 8, for example, then stop at that number.



Approach used :

- Generate ith permutation of an array without calculating the previous permutation using factroid numbers.
- Fix 0 as the starting point. Generate all (n-1)! Permutations of array. Find min cost path and update.