Project Report

Timed on MacBook Air, 2022, Apple Silicon M2, 8 GB memory, results may vary across devices

Nearest Neighbor:

Country	# of Cities	Length of Tour	Duration
Burma	33,708 Cities	0.30241e+06	25.305 seconds
Morocco	14,185 Cities	433,694	4.794 seconds
Yemen	7,663 Cities	199,429	0.924 seconds
Egypt	7,146 Cities	190,254	0.983 seconds
Argentina	9,152 Cities	0.20866e+06	1.869 seconds
Uruguay	734 Cities	90,246	.010 seconds
Western Sahara	29 Cities	30,246	0 seconds

Optimal Tour from Website:

Country	# of Cities	Length of Tour	Duration
Burma	33,708 Cities	959,289	24 hours
Morocco	14,185 Cities	427,378	10 hours
Yemen	7,663 Cities	238,314	236,494 seconds
Egypt	7,146 Cities	172,386	46,249 seconds
Argentina	9,152 Cities	837,479	24, 301 seconds
Uruguay	734 Cities	79,114	3,507.21 seconds
Western Sahara	29 Cities	27,603	0.09 seconds

Discussion:

While my Nearest Neighbor algorithm excels in producing timely tour estimations for larger inputs, it falls short of achieving optimal solutions. The Argentina dataset also produced a very high tour length but after comparing it with optimal tours, it was clear it was because of Argentina's geography - countries with the most cities, supported by the map, are often more dense. The Nearest Neighbor algorithm proves particularly suitable for efficient, approximate solutions in cases with approximately 10,000 cities. Although Argentina serves as an exception, the algorithm generally delivers satisfactory estimations for most datasets. The search is O(N) for each iteration of the loop. The loop itself runs n times (where n is the number of nodes); thereafter the overall complexity of the loop is O(n^2).