Project

Due: 11:59 PM, May 15

Implement any one of the three problems below in Python using mpi4py. Submit the code of your program and a brief report. The report should describe the design of your program, and demonstrate the results of running your program.

1. **n-body solver**. Implement a parallel n-body solver with n = 10 particles. You may randomly generate the masses, initial velocities, and initial positions of all particles at the beginning. Compute the positions of all particles after 1 second. Compare the results by using 0.1s, 0.01s, and 0.001s as the time in each timestep, respectively.

Note: You may get 80% of the full grade if your code follows the basic solver (Chap. 6.1.9) or 100% if your code follows the reduced solver (Chap. 6.1.10, 1st edition).

2. **tree search**. Solve the TSP problem with n = 10 cities. You may randomly generate the costs between each pair of cities. Note that the costs between any two cities may be different in two directions. Show the best tour and its cost.

Note: You may get 80% of the full grade if your code follows the static portioning (Chap. 6.2.11) or 100% if your code follows the dynamic partitioning (Chap. 6.2.12, 1st edition).

3. **sample sort**. Use sample sort to sort 10000 randomly generated integers in parallel. Compare the runtime with different numbers of processes (e.g., 2/4/8).

Note: You may get 80% of the full grade if your code follows the first implementation or 100% if your code follows the second implementation (both in Chap. 7.2.8, 2nd edition).