INSTITUTE OF COMPUTER SCIENCE
UNIVERSITY OF THE PHILIPPINES LOS BAÑOS
College of Arts and Sciences, University of of the Philippines Los Baños
4031 College, Laguna, Philippines
Tel: (049) 536 2313 Fax: (049) 536 2302

CMSC 180: Introduction to Parallel Computing
Second Semester AY 2020-2021

Project Specifications

Kristine Bernadette P. Pelaez

Overview

The project is a synthesis of all of the things that have been discussed in the course. You are to implement serial and parallel algorithms for certain algorithms, analyze and explain how they are implemented, and how their respective implementations affected their running times. Also, as usual, performance should be measured and compared with each other.

Instructions

There will be four (4) different sets for the project. The sets will only differ on what algorithms you will implement, but the analysis and guide questions are mostly the same. **Your laboratory instructor will choose which set you will be doing.**

Implementations

You are required to implement the serial and parallel version of your assigned algorithms. If there are multiple serial versions for your algorithm, try to implement the best implementation; but if it requires additional concepts not covered in your previous CMSC courses, whatever was taught to you before would do[1].

For the parallel implementations, you can research possible implementations or create your own parallel implementation. In both cases, you must explain the decomposition technique used, the parallel model (if applicable), and the algorithm itself. And as per usual, references should be properly cited.

Do not print the output of your algorithm. But for testing the correctness of your implementation, create a function that prints the output of your algorithm. The function should be able to handle different values of N.

Performance Evaluation

The maximum size of the input data, N, depends on what your machine can handle. You are to test each algorithm starting from N=1000, incrementing by 1000, and stopping only when the run time of the current algorithm is equal or exceeds 1 hour.

Make sure to use the parallel algorithm metrics discussed to compare the performance of your implementations.

[1] Still, researching about the algorithms is encouraged.

INSTITUTE OF COMPUTER SCIENCE
UNIVERSITY OF THE PHILIPPINES LOS BAÑOS
College of Arts and Sciences, University of of the Philippines Los Baños
4031 College, Laguna, Philippines
Tel: (049) 536 2313 Fax: (049) 536 2302

CMSC 180: Introduction to Parallel Computing
Second Semester AY 2020-2021

Project Sets

Set A - Sorting Algorithms

Algorithms

Compare the serial and parallel versions of the following sorting algorithms:

1. Bubble sort
2. Merge sort
3. Shell sort
4. Bucket sort

The output of each algorithm should be the sorted array.

Testing

Randomly generate a one-dimensional array of size N. Values in the array should range from $[-\frac{N}{2}, \frac{N}{2}]$, repetitions are allowed.

Set B - Graph Algorithms

Algorithms

Compare the serial and parallel versions of the following graph algorithms:

1. Prim's Algorithm (for Finding the Minimum Spanning Tree of a Graph)
2. Dijkstra's Algorithm (for Single-source Shortest Path)

The output of both algorithms should be the edges included in the MST or shortest path.

Testing

Randomly generate an undirected graph with N vertices. There should be at least $\frac{N^2}{2}$ edges in the graph. Each edge in the graph has a weight of 1. It is up to you whether you will represent your graph via an adjacency matrix or an adjacency list.

Before running your algorithm, check first if your graph is connected, i.e., there is a path from one vertex to every other vertex in the graph.

For the Prim's Algorithm Randomly choose a vertex from the N vertices where the Prim's Algorithm will start looking for the MST.

For the Dijkstra's Algorithm Randomly choose two vertices, one would be the source vertex while the other is the destination vertex.

Set C - Search Algorithms

Algorithms

Compare the serial and parallel versions of the following search algorithms:

1. Binary Search Algorithm (in a one-dimensional array of size N)
2. Depth-First Search Algorithm (for finding path from one vertex to another)

The output of the binary search algorithm should be the value being search and its index. For the DFS Algorithm, the edges of the path should be printed.

CMSC 180: Introduction to Parallel Computing
Second Semester AY 2020-2021

Testing

For the Binary Search Algorithm Generate a one-dimensional array of size N. Values in the array should range from $[-\frac{N}{2}, \frac{N}{2}]$, repetitions allowed.

Randomly generate a number to search in the array. You *do not* need to guarantee that the number exists in the array, but make sure that the algorithm catches it when the number is not in the array.

For the Depth-First Search Algorithm Randomly generate an undirected graph with N vertices. There should be at least $\frac{N^2}{2}$ edges in the graph. It is up to you whether you will represent your graph via an adjacency list or adjacency matrix.

You do not need to check if the graph generated is connected or not, but your algorithm must detect when there is no path from one vertex to another.

Set D - Parallel Dynamic Programming

Algorithm

Compare the serial and parallel implementation of the algorithm below:

1. 0-1 Knapsack Problem

The output should be the indices of the items to be packed and its corresponding weights.

Testing

Randomly generate N items, each with random weights ranging from [1, 100]. These items will be the items to choose from. Next, randomly generate a value from $[\frac{N}{2}$, N] which will be the capacity of the knapsack.

Requirements

You are required to submit the following:

Code

Same as your previous exercises, you are to submit your implementations for the algorithms assigned to you. However, the src folder should be organized into subdirectories, one for each algorithm you will implement. For instance, if you are to implement algorithm A, then you must have a folder named algorithm A which contains your serial and parallel implementation for that algorithm.

Lab Report

The lab report should be uploaded in the Google Classroom assignment post. It should contain your setup, overview of the algorithms you implemented, performance evaluation, and analysis of the algorithm and the resources consumed (theoretical and actual).

Since there will be no handout given for these algorithms, you are **required** to cite your sources, no matter how trivial the contribution is to your actual paper. Citations should be in-text, and should be properly listed at the end of the document.

INSTITUTE OF COMPUTER SCIENCE
UNIVERSITY OF THE PHILIPPINES LOS BAÑOS
College of Arts and Sciences, University of of the Philippines Los Baños
4031 College, Laguna, Philippines
Tel: (049) 536 2313 Fax: (049) 536 2302

CMSC 180: Introduction to Parallel Computing
Second Semester AY 2020-2021

Guide Questions

The contents of your paper for the project should be almost the same with the previous exercises, if not having more explanations and comparisons.

Here are some guide questions and reminders for your project report:

1. What are the theoretical running times of your implementations? Did the actual running times conform with the theoretical running times?
2. Analyze the space consumption of your algorithms. Did they affect the running time of your implementations? How?
3. Compare and contrast the running time of each algorithm. Which is the *best* algorithm to use? Consider possible restrictions to resources. Make sure to list the data (if there are a lot of data, you can add an appendix page for your tables) and have visual comparisons for your data.
4. What are the advantages/disadvantages of each of your implementation, if any? Explain why they are advantages/disadvantages.
5. Is implementing the parallel version of the algorithms worth it? Explain.

Deliverables

See the Lab Guide for the deliverables.

Scoring

- Code: **35 points**
- Lab Report: **40 points**

Questions?

If you have any questions or comments regarding this material, please contact your instructor.