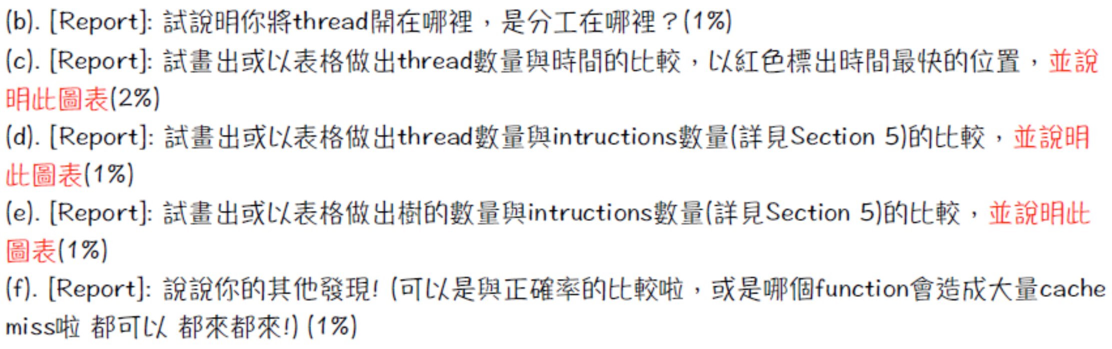
Name: Calvin Liu (劉益瑋)
Student ID: B06902100

Systems Programming Homework 4 Report

Questions



(b). [Report]: 試說明你將thread開在哪裡，是分工在哪裡？(1%)

(c). [Report]: 試畫出或以表格做出thread數量與時間的比較，以紅色標出時間最快的位置，並說明此圖表(2%)

(d). [Report]: 試畫出或以表格做出thread數量與intructions數量(詳見Section 5)的比較，並說明此圖表(1%)

(e). [Report]: 試畫出或以表格做出樹的數量與intructions數量(詳見Section 5)的比較，並說明此圖表(1%)

(f). [Report]: 說說你的其他發現! (可以是與正確率的比較啦，或是哪個function會造成大量cache miss啦 都可以 都來都來!) (1%)

Answers (in English)

1. **Thread Creation and Work Distribution**

   The threads were created before the training and testing steps. All threads other than the main thread would terminate after each step.

   In the **training process**, the threads are in charge of the creation of a tree. A thread would check if there were any trees left to create, and if there were, it would plant a tree and store its root into an array. If there were not (no more trees left to create), it would terminate.

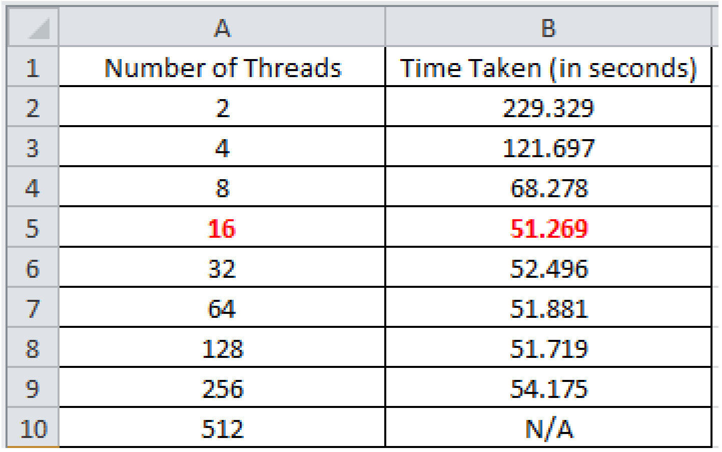
   For the **testing process**, each thread is tasked to handle the prediction of an input. It would first check if there were any input(s) left to process, and would terminate when there were none left.

2. **Number of Threads and Time Comparison**

   To do this comparison, we will set two values as constants. We will set the number of trees = 3000 and the amount of entries during the planting of a tree = 128. This is tested in workstation.

   Why 3000: to make the differences in time more noticeable.

   *the submission file will lower the number of trees planted to 2000 (no difference in accuracy).

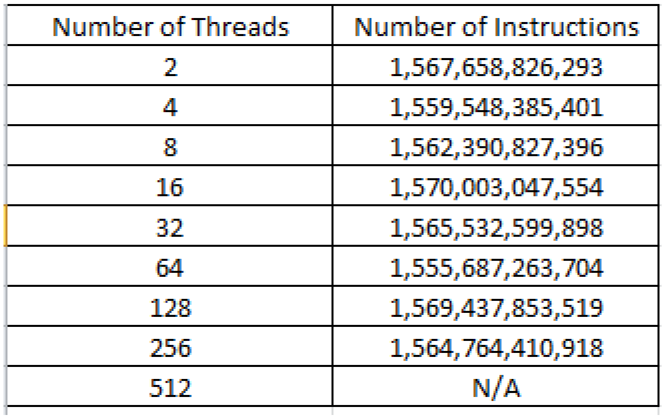


	A	B
1	Number of Threads	Time Taken (in seconds)
2	2	229.329
3	4	121.697
4	8	68.278
5	16	51.269
6	32	52.496
7	64	51.881
8	128	51.719
9	256	54.175
10	512	N/A

Based from the table, presumably the time stabilizes after 16 threads (or around that). Adding more threads (N.o.T. ≥ 16) makes little to no difference. The time is not available when Number of Threads = 512 since the workstation can not handle it.

3. Number of Threads and Instructions Comparison

Like previous section, we set number of trees = 3000 and the amount of entries during the planting of a tree = 128.

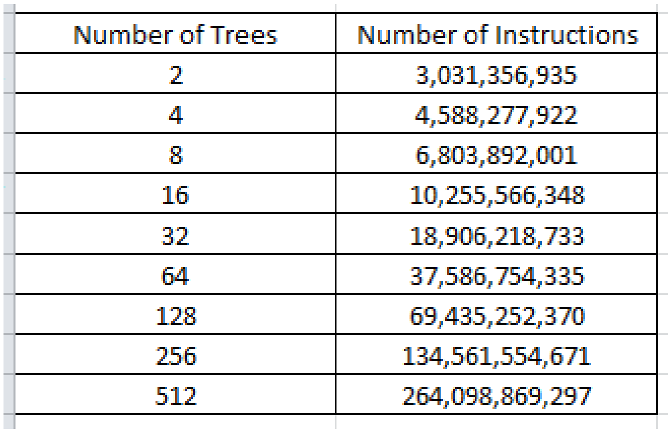


Number of Threads	Number of Instructions
2	1,567,658,826,293
4	1,559,548,385,401
8	1,562,390,827,396
16	1,570,003,047,554
32	1,565,532,599,898
64	1,555,687,263,704
128	1,569,437,853,519
256	1,564,764,410,918
512	N/A

Based from the data of the table, it seems that there are no visible correlation between the amount of threads and the number of instructions used.

4. Number of Trees and Instructions Comparison

For this section, we set the number of threads = 16 and the amount of entries during the planting of a tree = 128.



Number of Trees	Number of Instructions
2	3,031,356,935
4	4,588,277,922
8	6,803,892,001
16	10,255,566,348
32	18,906,218,733
64	37,586,754,335
128	69,435,252,370
256	134,561,554,671
512	264,098,869,297

Based from the data of the table, we can see that it seems that there is a direct correlation between the number of trees and the number of instructions called. (more trees, more instructions)

5. Other Discoveries

- The efficiency of threads vary greatly between systems. (amounts of parallelization)
- On the workstation, creating 512 or more threads would result in the program never get finished.
- The accuracy stabilized at around 89% (no matter how many trees planted, the accuracy never reached 90%).

*NOTE

I used tables to represent my data since it's the easiest way to do. If it's not sufficient, please notify me.