CSCI-UA.0480-051: Parallel Computing
Midterm Exam (Oct 17th, 2024)
Total: 100 points

Important Notes- READ BEFORE SOLVING THE EXAM

- **If you perceive any ambiguity in any of the questions, state your assumptions clearly and solve the problem based on your assumptions. We will grade both your solutions and your assumptions.**
- **This exam is take-home.**
- **The exam is posted on Brightspace, at the beginning of the Oct 17th lecture (2pm EST).**
- **You have up to 24 hours to submit on Brightspace (i.e. till Oct 18th 2pm EST), in the same way as you submit an assignment. However, unlike assignments, you can only submit once.**
- **Your answers must be very focused. You may be penalized for giving wrong answers and for putting irrelevant information in your answers.**
- **Your answer sheet must be organized as follows:**
  - **The very first page of your answer must contain only:**
    - **You Last Name**
    - **Your First Name**
    - **Your NetID**
    - **Copy and paste the honor code shown in the rectangle at the bottom of this page.**
  - **In your answer sheet, answer one problem per page. The exam has three main problems, each one must be answered in a separate page. If your solution for a problem takes more than a page, that is fine, but the following program must start on a new page.**
- **This exam consists of 3 problems, with a total of 100 points.**
- **Your answers can be typed or written by hand (but with clear handwriting). It is up to you. But you must upload one pdf file containing all your answers.**

Honor code (copy and paste to the first page of your exam)

- You may use the textbook, slides, the class recorded lectures, the information in the discussion forums of the class on Brightspace, and any notes you have written.
- You may NOT use communication tools to collaborate with other humans. This includes but is not limited to Google-Chat, Messenger, E-mail, etc.
- You cannot use LLMs such as ChatGPT, Gemini, etc.
- Do not try to search for answers on the internet, it will show in your answer, and you will earn an immediate grade of 0.
- Anyone found sharing answers, communicating with another student, searching the internet, or using prohibited tools (as mentioned above) during the exam period will earn an immediate grade of 0.
- **"I understand the ground rules and agree to abide by them. I will not share answers or assist another student during this exam, nor will I seek assistance from another student or attempt to view their answers."**

Problem 1

a. [5 points] Does a hyperthreading core need to be also pipelined? Justify your answer.

b. [5 points] Suppose we have a program that has three parts. The first part, 30% of the code execution, is totally sequential. The second part, 30% of the code execution, can be parallelized using four threads executing on four cores. The last part, 40% of the code execution, can only be parallelized using two threads executing on two cores. Suppose we have a processor with only four physical cores, each one is just pipelined. What is the speedup we can get? Assume that the first part (the sequential) must be executed first, then second and third parts (i.e. the parts that can be parallelized) can execute in any order (i.e. second part then third part, or the other way around). Show all the steps to get full credit. A correct final answer without steps will not earn any credit.

c. [5 points] Repeat problem "b" but using a processor with eight cores each of them is a pipelined one.

d. [10 points] Assume we have three cores c1, c2, and c3. Each core has its own private cache. Each of the cores accesses address A as indicated below. Assume the caches initially do not contain the cache block that contains data in address A. The coherence protocol used is MESI directory-based write-invalidate. All caches are write-through. Fill out the table below that shows the state of the cache block that contains the data in address A in each cache after the operation of each row. (e.g. at time 1, what will be status of "block A" in the cache of c1, c2, and c3 *after* c1 reads block A?)

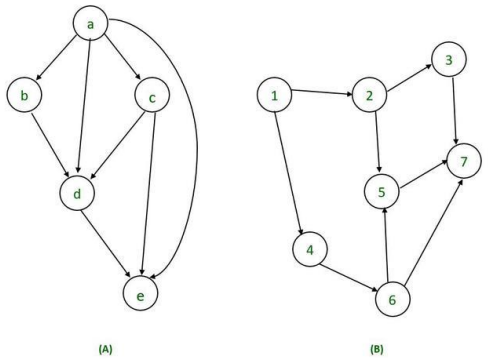
time	Operation	Status of Block A at Cache of C1	Status of Block A at Cache of C2	Status of Block A at Cache of C3
1	c1 reads block A			
2	c2 reads block A			
3	c3 writes block A			
4	c1 reads block A			
5	c2 writes block A			
6	c3 reads block A			
7	c3 writes block A			

[Note: "block A" in the table above means the block that contains data in address A].

e. [10 points] Repeat problem "d" above if the coherence protocol is MESI but snoopy write-invalidate.

Problem 2

Suppose we have the following two DAGs each of which represents a possible execution of an algorithm. Also suppose each node in DAG A takes two units of time to execute while each node of DAG B takes only one unit of time.



a. [5 points] If you are given as many cores as you want, which of the two DAGs, A and B, is better in terms of speedup? Show detailed steps that made you reach the conclusion. If they are both equal, also show the steps.

b. Suppose you can adjust the algorithm design and result in removing an edge from DAG A. Also assume you have as many cores as you want.

1. [5 points] Which edge, if any, to remove from DAG A to keep the speedup you calculated in part "a" above fixed? That is, if we remove this edge, the speedup you calculated stays the same. If there is more than one solution, state them all. If there is none, say so.
2. [5 points] Which edge, if any, to remove from DAG A to increase the speedup you calculated in part "a"? That is, if we remove this edge, the speedup you calculated increases. And calculate the new speedup. If there is more than one solution, state them all. If there is none, say so.

c. [5 points] For DAG A, which is better: to have four physical cores, each one is pipelined only, or one core with four-way hyperthreading? Justify your answer. Assume different nodes require different types of arithmetic operation.

d. [5 points] If you can combine two or more nodes from DAG B to increase the speedup from the one you calculated in part "a" above, which ones you will combine? And what is the new speedup? If there is more than one solution, state them all. If there is no solution, state so too. [Note: When you combine more than one node, the new bigger node takes the sum of the times of all the nodes used to form this node. For example, if you combine two nodes each one takes one unit of time, the new bigger node takes two units of time.]

d. [10 points] Suppose we have the following piece of code; can it be represented by DAG A? or DAG B? Explain your answer by showing which instruction map to which node. Each node can represent one or more instructions from the ones below. If the code does not map to any of the two DAGs, say so and draw the new DAG that represents the code below with one node per instruction.

```
x= 5;
y = x*x;
z = x+x;
k = y++;
l = y*h;
h = z*4;
m = l + h + k;
```

Problem 3

a. [5 points] Suppose we execute the *same* piece of code on two different multicore processors: A and B. We found that the efficiency of processor A when executing the code is 100% while the efficiency of processor B when executing the code is 80%. Does this mean that the code will surely execute faster on processor A than B? Justify.
[Assume Efficiency = speedup / #cores].

b. [10 points] The following piece of code cannot be parallelized with loop-level parallelism (i.e. executing several iterations in parallel) due to loop carried dependencies. Is there a way to modify the code to be parallelizable? If yes, show the new parallelizable version. If not, explain why not.

```
for (int i = 1; i < N; i++)
{
 A[i] = B[i] – A[i – 1];
}
```

c. Suppose a thread is executed on a pipelined core (not superscalar or hyperthreading).
The clock frequency of the core is 2 GHz. The instructions of the thread are as follows:

Instruction type	**#instructions (in millions)**	**#cycles per instruction**
add/sub	5	2
Mult/div	2	10
branches	1	5

1. [5 points] What is the MIPS of executing this thread on the core? Show all your steps.
2. [5 points] What is the CPI of executing this thread on the core? Show all your steps.
3. [5 points] What is the total execution time of executing this thread on the core? Show all your steps.