Problem 1: Potpourri (50 pts)

A) [8 pts] Amdahl's Law

What assumption is made by Amdahl's Law about the parallelizable fraction of a program?

That it is perfectly parallelizable.

What are the three major reasons why this assumption may not hold?

1) **Synchronization overhead**

2) **Load imbalance overhead**

3) **Resource sharing overhead**

B) [9 pts] Locking

Give three reasons why a lock may be required statically for program correctness but may not be needed dynamically?

1) **Threads may not update the shared data protected by the lock.**

2) **Threads may update disjoint parts of the shared data structure protected by the lock.**

3) **Threads may not contend for the lock.**

C) [10 pts] Memory Consistency

Consider the following statement:

"A sequentially consistent multiprocessor guarantees that different executions of the same multithreaded program produce the same architecturally-exposed ordering of memory operations."

1) Is this statement true or false?

CIRCLE ONE: **TRUE** **FALSE**

2) Explain your reasoning (less than 15 words).

Sequential consistency makes no guarantees across different executions. (It is about the ordering of operations within the *same* execution)

3) Why do we want the property described above; i.e., the property that "different executions of the same multithreaded program produce the same architecturally-exposed ordering of memory operations"?

Debugging ease.

D) [9 pts] SLE vs. TM

1) What is the major difference between speculative lock elision (SLE) and transactional memory (TM)?

TM requires the programmer to mark transactions. SLE preserves conventional lock based programming.

2) What benefit does TM provide that SLE does not?

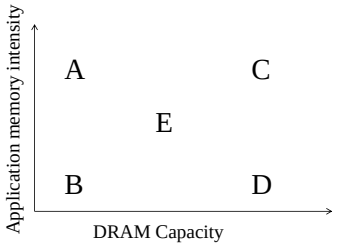
No need for reasoning about locks and getting them correct.

3) What benefit does SLE provide that TM does not?

No need to modify existing lock-based program.

E) [14 pts] A Refreshing Problem

Recall from lecture that RAIDR is a mechanism that uses profiled DRAM cell retention times to identify and skip unnecessary refreshes. Below is a plot with two independent variables—application memory intensity and DRAM capacity.



1) Identify the point (A, B, C, D, or E) where a mechanism like RAIDR would buy the most performance relative to a system without RAIDR.

CIRCLE ONE: **A** **B** **C** **D** **E**

CIRCLE ONE: **A** **B** **[C]** **D** **E**

Why (15 words or less)?

There is much DRAM to refresh and many memory accesses with which refreshes would contend.

2) Identify the point (A, B, C, D, or E) where the most energy is spent on refreshes (relative to total DRAM energy).

CIRCLE ONE: **A** **B** **C** **D** **E**

CIRCLE ONE: **A** **B** **C** **[D]** **E**

Why (15 words or less)?

There is much DRAM to refresh and few memory accesses, so refresh energy dominates.