**Problem 4: Tiered-difficulty** (30 pts)

Recall from your required reading on Tiered-Latency DRAM that there is a near and far segment, each containing some number of rows. Assume a very simplified memory model where there is just one bank and there are two rows in the near segment and four rows in the far segment. The time to activate and precharge a row is 25ns in the near segment and 50ns in the far segment. The time from start of activation to reading data is 10ns in the near segment and 15ns in the far segment. All other timings are negligible for this problem. Given the following memory request stream, determine the optimal assignment (minimize average latency of requests) of rows in the near and far segment (assume a fixed mapping where rows cannot migrate, a closed-row policy, and the far segment is inclusive).

```
time 0ns:  row 0 read
time 10ns:  row 1 read
time 100ns:  row 2 read
time 105ns:  row 1 read
time 200ns:  row 3 read
time 300ns:  row 1 read
```

**Detailed solution**

**If you were to map 0 and 2 (this is the answer) to near segment:**

```
row 0:  activated at time = 0
row 0:  read at time = 10 (10ns latency)
row 1:  activated at time = 25
row 1:  read at time = 40 (30ns latency)
row 2:  activated at time = 100
row 2:  read at time = 110 (10ns latency)
row 1:  activated at time = 125
row 1:  read at time = 140 (35ns latency)
row 3:  activated at time = 200
row 3:  read at time = 215 (15ns latency)
row 1:  activated at time = 300
row 1:  read at time = 315 (15 ns latency)
```

**total latency is 115ns**

**If you were to map 1 and 2 (an example incorrect answer) to near segment:**

```
row 0:  activated at time = 0
row 0:  read at time = 15 (15ns latency)
row 1:  activated at time = 50
row 1:  read at time = 60 (50ns latency)
row 2:  activated at time = 100
row 2:  read at time = 110 (10ns latency)
row 1:  activated at time = 125
row 1:  read at time = 135 (30ns latency)
row 3:  activated at time = 200
row 3:  read at time = 215 (15ns latency)
row 1:  activated at time = 300
row 1:  read at time = 310 (10 ns latency)
```

**total latency is 130ns**

**A) [6 pts]** What rows would you place in near segment? Hint: draw a timeline.

rows 0 and 2. see above

**B) [6 pts]** What rows would you place in far segment?

rows 1 and 3 (also rows 0 and 2 since inclusive). see above

**C) [6 pts]** In 15 words or less, describe the insight in your mapping?

See TL-DRAM's WMC policy – the first access in near simultaneous requests causes the second to wait activation + precharge time. minimizing this wait by caching first row in near segment is better than caching second row in near segment (this decreases only time to read from start of activation), even if second row is accessed more frequently (see example above)

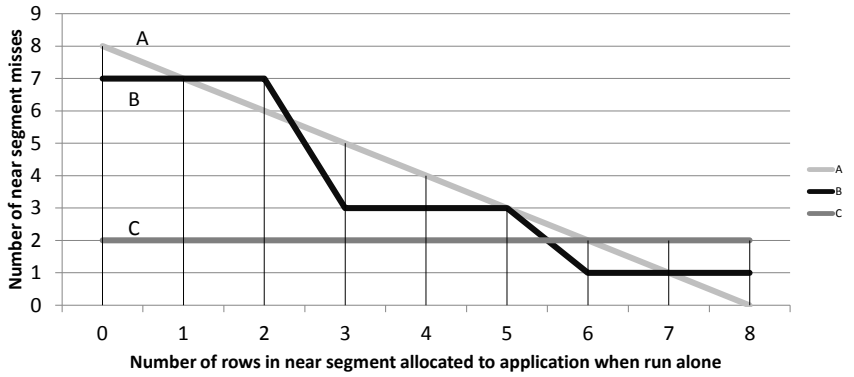
**D) [6 pts]** Assume now that the mapping is dynamic. What are the tradeoffs of an exclusive design vs. an inclusive design? Name one advantage and one disadvantage for each.

Exclusive requires swapping, but can use nearly full capacity of DRAM. Inclusive, the opposite.



Continued . . .

**E) [6 pts]** Assume now that there are eight (8) rows in the near segment. Below is a plot showing the number of misses to the near segment for three applications (A, B, and C) when run alone with the specified number of rows allocated to the application in the near segment. This is similar to the plots you saw in your Utility-Based Cache Partitioning reading except for TL-DRAM instead of a cache. Determine the optimal static partitioning of the near segment when all three of these applications are run together on the system. In other words, how many rows would you allocate for each application? Hint: this should sum to eight. Optimal for this problem is defined as minimizing total misses across all applications.




**1)** How many near segment rows would you allocate to A?

| 5 |
|---|

**2)** How many near segment rows would you allocate to B?

| 3 |
|---|

**3)** How many near segment rows would you allocate to C?

| 0 |
|---|