

# 18-344: Computer Systems and the Hardware-Software Interface

## HW03: Caches

**Due:** Thursday, Sept 29th, 2022 at 10 am

Name: \_\_\_\_\_

Andrew ID: \_\_\_\_\_

### Homework instructions:

These homeworks are designed be to exercises for you to check your understanding of concepts as we go through lectures and to prepare you for the kinds of questions that you may see on the midterm. They are not intended to take a significant amount of time. If you find that you have spent more than an 1 hour on the homework, note your progress down, write your best guess and then come to office hours. You will be given credit for any good faith attempt.

### How to submit homeworks:

Download from this file from the course webpage and edit it with whatever PDF editor you're most comfortable with. Some recommendations from other students and courses that use Gradescope include:

**pdfescape.com**                      A web-based PDF editor that works on most, if not all, devices.

**Preview**                                Pre-installed default MacOS PDF Editor.

**iAnnotate**                              A cross-platform editor for mobile devices (iOS/Android).

If you'd prefer not to edit a PDF, you can print the document, write your answers in neatly and scan it as a PDF. Once you've completed the lab, upload and submit it to *Gradescope*.

Question:	1	2	3	Total
Points:	4	6	10	20
Score:				

## 1. Locality, Locality, Locality

Examine the following implementation of matrix multiply. (Assume that memory is stored in row-major order).

```
1 for (i=0; i<n; i++) {  
2     for(j=0; j<n; j++) {  
3         float sum = 0.0;  
4         for (k=0; k<n k++) {  
5             sum += a[i][k] * b[k][j];  
6         }  
7         c[i][j] = sum;  
8     }  
9 }
```

2 pts

**1.1** identify each of the memory accesses which exemplify spatial locality, and mark their corresponding box. Do not mark boxes which do not exemplify spatial locality.

- i
- j
- k
- n
- sum
- a[i][k]
- b[k][j]
- c[i][j]

2 pts

**1.2** identify each of the memory accesses which exemplify temporal locality, and mark their corresponding box. Do not mark boxes which do not exemplify temporal locality.

- i
- j
- k
- n
- sum
- a[i][k]
- b[k][j]
- c[i][j]

## 2. ABC's of Caches

For each cache parameter identify how tuning that parameter impacts compulsory misses, conflict misses, and/or capacity misses if at all.

2 pts

### 2.1 Associativity

2 pts

### 2.2 Block Size

2 pts

### 2.3 Cache Size

### 3. Belady Optimize

Examine the following memory trace and compare the relationship between cache A and cache B. You may make the following assumptions:

- both caches are fully associative
- both caches have a block size of 1 byte
- both caches use byte addressable memory
- both caches are 4 bytes
- cache A uses a Last Recently Used replacement policy
- cache B uses a Belady Optimal replacement policy

10 pts

**3.1** Fill in the table for both Cache A and Cache B. The first two entries have been filled in as an example.

Access	Address	Cache A				Cache B			
1	0x3	0x3				0x3			
2	0x7	0x3	0x7			0x3	0x7		
3	0x3								
4	0x5								
5	0x9								
6	0xf								
7	0x3								
8	0x3								
9	0x7								
10	0x5								
11	0x6								
12	0x3								