Cpt S 260 Homework #7

Please print your name!

You may want to wait for me to cover the material later this week for next week.

0. I have a small book shelf on my bed, holding 10 books at most. I also have a large bookshelf in my studying room, having 1000 books exactly. Suppose that fetching a book from the small shelf takes 60 seconds, while fetching a book or more, in a batch, from the large shelf takes 5 minutes. Assume that the books on the large shelf is sorted alphabetically according to its title and the small shelf is the only place to hold books in my bedroom. Every night, I read one book, and at the beginning of the month, I choose randomly a book to read, and the books I read afterwards are also alphabetically ordered according to their titles. Tell me what is the minimal averge seconds needed to fetch a book.

1. We are given a direct-map cache with 1024 blocks, where each block is a MIPS word (4 bytes). The cache uses write-back when a write miss happens. The main memory is of 2^{30} words. Initially, the cache is empty. For the following sequences of instructions, determine the number of cache misses and give some reasoning.

(1).

```
lw $s0, 8($sp)
lw $s0, 12($sp)
sw $s0, 8($sp)
sw $s1, 8($sp)
lw $s0, 8($sp)
```

(2).

```
lb $s0, 8($sp)
lb $s0, 12($sp)
sb $s0, 8($sp)
sb $s1, 8($sp)
lb $s0, 8($sp)
```

2. We are given a cache with the following parameters. For the instruction cache, its miss rate is 7%, and its miss penalty is 100 cycles. For the data

cache, its miss rate is 5%, and its miss penalty is 200 cycles. Assume that a typical sequence of instructions has CPI = 2, and contains 30% memory accesses. Compute the ratio of improvement in performance if this cache is replaced by a perfect cache (which never misses).

3. Consider a 4-way set associative cache of size 8 (blocks). This cache has two sets: set0 and set1. The cache uses the least recently used (LRU) scheme for block replacement on cache misses. Initially, the cache is empty. Show the contents of the cache at each step of the following sequence of memory accesses:

a. block00
b. block04
c. block02
d. block08
e. block04
f. block12
g. block04
h. block08