Homework 7

Jarred Parr

1. a. The lower bound of page faults will be the size of the window which, in this case, is *m*.

   b. The upper bound of page faults would occur in the situation that all of the window entries are always missing the next value in the reference string. So the upper bound is *p*.

2. a. FIFO = 14

   b. Optimal = 8

   c. LRU = 10

3. a. `32 - 12 = 20`

p	d
20	12

   b. `22 - 12 - 10`

f	d
20	12

   c. `0100 0101 0110`

4. `0.8(2) + 0.18(2) + 0.02(20000 + 2)` = 401.2 msec

Practical

```
if (referenced_ptes) {
    if (PageSwapBacked(page))
        return PAGEREF_ACTIVATE;
    /*
     * All mapped pages start out with page table
     * references from the instantiating fault, so we need
     * to look twice if a mapped file page is used more
     * than once.
     *
     * Mark it and spare it for another trip around the
     * inactive list.  Another page table reference will
     * lead to its activation.
     *
     * Note: the mark is set for activated pages as well
     * so that recently deactivated but used pages are
     * quickly recovered.
```

```
     */
    SetPageReferenced(page);

    if (referenced_page || referenced_ptes > 1)
        return PAGEREF_ACTIVATE;

    /*
     * Activate file-backed executable pages after first usage.
     */
    if (vm_flags & VM_EXEC)
        return PAGEREF_ACTIVATE;

    return PAGEREF_KEEP;
}
```

This page references what happens when a page is referenced and is soon to be referenced again, it will keep it in the frame so that way it will incur fewer faults which occurs when reloading the data from memory. This is similar to the optimal cache without needing to quite know in advance the entire reference string. If it knows that the piece of data currently in the frame is being used, then it will keep it in the frame for the next pass.