Problem 1 (0 Points)
Consider a buddy system and the address 011011110000.

a) If the block size associated with this address is 8 bytes, what is the binary address of the buddy?

b) What is the largest block size $N$ such that the above address still has a buddy? Explain.

Problem 2 (4 Points)
Consider a buddy system and the address 100100001000. Assume the largest block has $2^U$ bytes and the smallest block has $2^L$ bytes.

a) If the block size is 8 bytes, what is the binary address of the buddy?

b) What is the largest block size $N$ such that the above address still has a buddy? Explain.

c) Let $b_k(x)$ be the buddy of address $x$ with block size $2^k$. Write an expression for $b_k(x)$. Explain why the form of the expression is correct. If you can’t write the expression, then give the algorithm for computing $b_k(x)$.

d) Demonstrate that your expression in Part c is correct.

Problem 3 (0 Points)
Consider the following page reference sequence:

\[
1, 3, 3, 3, 3, 3, 3, 1, 1, 0, 3, 3, 3, 2, 4, 2, 3, 3, 2, 3, 3, 2, 2, 2
\]

Suppose that the process is allocated 3 frames and that all frames are initially empty.

a) Which page references will cause a page fault if the LRU page replacement algorithm is used?

b) What will be the page fault rate if an LRU page replacement algorithm is used?
Problem 4 (5 Points)
Suppose that a process has been allocated 4 page frames (numbered 0 to 3) and initially (time = 0), frame k contains virtual page k (e.g., frame 2 contains page 2). Furthermore, suppose that the following is the reference string (RS) where the memory operation is indicated by a 'r' (read) or 'w' (write):

\[ [0r \ 1r \ 2r \ 3r] \ 2r \ 0w \ 3r \ 1w \ 4r \ 1r \ 0w \ 1r \ 2r \ 3r \]

where the bracketed references were due to the initial loading phase. For the page replacement algorithms below, fill out the table below which shows the evolution of the page table:

<table>
<thead>
<tr>
<th>Time T</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>RS</td>
<td>2r</td>
<td>0w</td>
<td>3r</td>
<td>1w</td>
<td>4r</td>
<td>1r</td>
<td>0w</td>
<td>1r</td>
<td>2r</td>
<td>3r</td>
<td></td>
</tr>
</tbody>
</table>

Assume that initially all R-bits and M-bits are 0 and that the clock cursor is pointing at frame 0. Also assume that

a) Optimal; b) FIFO; c) LRU; d) Clock; e) NRU.

In the case of the Optimal algorithm, assume also that the reference sequence for the time period from T=11 to T=14 is 0r, 1r, 2r, 3r.

Problem 5 (0 Points)
Problem 26, Chapter 4, Tanenbaum.

Problem 6 (0 Points)
Problem 27, Chapter 4, Tanenbaum.

Problem 7 (4 Points)
Problem 25, Chapter 4, Tanenbaum.
**Problem 8 (6 Points)**

Consider the working set model with a window size of $\Delta = 3$ and the following page reference sequence for a process:

$$3r, 2r, 2r, 2r; 2r, 2r, 3r, 3w, 0r; 2r, 2r, 3r, 4r; 3w, 4r, 1r, 4w; 2r, 2r, 1r, 1r, 1r$$

a) What is the working set for $t = 5, 10, 15, 20$ and $25$? Note that these times correspond to the semicolons in the page reference sequence. Assume that the working set is computed after each page reference.

b) It is impractical to compute the working set after each page reference. Section 4.4.9 describes WSClock, a more practical algorithm. Give an example of how that algorithm would work here if the periodic clock interrupt occurred every 5 page references above; i.e., at the location of the semicolons in the page reference sequence. Assume the following:

- $\tau$ is 2 ticks.
- Initially, pages 2 and 3 are the only pages in memory ($V$-bit = 1) and their R- and M-bits are 0. Tanenbaum refers to the V(alid)-bit as the *Present/Absent* bit.
- You can assume that the clock starts at $T = 0$ and is incremented at each clock interrupt.

To demonstrate your answer, you will need to show how the $V$- and R-bits evolve at each page fault.