CS1704 Final Exam Koofer

I. Searching

Give the Orders of the following search methods.

Sequential Searching:

Average Case: $O\left( \frac{N}{2} \right)$

Worst Case: $O\left( N \right)$

Sequential Searching with Probability Ordering:

Worst Case: $O\left( N \right)$

Binary Searching:

Worst Case: $O\left( \log_2 N \right)$
II. Sorting

Below are the contents of an unsorted array of integers:

1  2  3  4  5  6  7  8  9  10
39 58 17 51 75 24 56 91 33 84

Fill in the array diagrams below to reflect the contents of the array after each of the first two passes using the designated sorting algorithm. Assume the algorithms are sorting the array elements in ‘descending’ order.

**Duplex-Selection Sort**: Pass 1

1  2  3  4  5  6  7  8  9  10
17 58 39 51 75 24 56 84 33 91

**Duplex-Selection Sort**: Pass 2

1  2  3  4  5  6  7  8  9  10
17 24 39 51 75 58 56 33 84 91

**Bubble Sort**: Pass 1

1  2  3  4  5  6  7  8  9  10
39 17 51 58 24 56 75 33 84 91

**Bubble Sort**: Pass 2

1  2  3  4  5  6  7  8  9  10
17 39 51 24 56 58 33 75 84 91
IV. Path Testing

```c
if (c <> 0) && (b <> 0)
    c = 7;
    a = 0;
    do
        if (c + c < b)
            b = b - c;
            a = a + 1;
        b = b - c;
        a = a + 1;
        while (b > c)
    cout << a;
```
Let the darkened circle numbers represent the Flow Graph node labels.

1. Identify the integer test value for \( b \) to check this code by performing a Reverse Path Analysis on the path:

\[ 1 , 2 , 3 , 4 , 5 , 6 , 3 , 5 , 6 , 7 \]  

\[ 21 \leq b < 28 \]  

(5 points)

2. List the conditions and restrictions that must be TRUE at the end of the Reverse Path Analysis.

\[
\begin{align*}
\text{c} &= 7 & \text{b} &\geq 3c \\
\text{c} &\neq 0 & \text{b} &< 4c \\
\text{b} &\neq 0
\end{align*}
\]

Reverse Path Analysis Trace:

#7  
\( \emptyset \)

#6  \( b < c \)  
\( \emptyset \)

#5  \( b-c < c \land b < 2c \)  
\( \emptyset \)

#3  \( b < 2c \land c + c > b \land b < 2c \)  
\( \emptyset \)

#6  \( b < 2c \land c \leq b \land c \leq b < 2c \)  
\( \emptyset \)

#5  \( c \leq b - c < 2c \land 2c < b < 3c \)  
\( \emptyset \)

#4  \( 2c < b - c < 3c \land 3c < b < 4c \)  
\( \emptyset \)

#3  \( 3c < b < 4c \land c + c < b \land 3c \leq b < 4c \land 2c < b \land 3c < b < 4c \)  
\( \emptyset \)

#2  \( 3c < b < 4c \land c \neq 0 \land b \neq 0 \)  
\( \emptyset \)

#1  \( 3c < b < 4c \land c \neq 0 \land b \neq 0 \land c = 7 \)
V. Data Type Design

Design the Data Type operations for reading a book. There is no need to discuss the physical representation of the implementation, (i.e. the actual data structure for storing the book). Simply list the operations that would be needed in a program for reading electronic books. In addition to specifying the operations, denote the parameters of the operations. Follow each operation, (procedure heading), declaration with one or two brief sentences describing the operation and the parameters. Do not specify unnecessary operations, (bells and whistles), a basic set of operations will be sufficient. Be complete and consistent.