

University of Nevada, Las Vegas Computer Science 477/677 Fall 2023

Assignment 3: Due Friday September 22 2023 11:59 PM

Upload your homework to Canvas.

Name: _____

You are permitted to work in groups, get help from others, read books, and use the internet. Your answers must be written in a pdf file and uploaded to canvas, by midnight September 2nd. Your file must not be unnecessarily long. If you have any questions, or you are having trouble uploading the assignment you may email the grader, Sebrina Wallace, at wallace4@unlv.nevada.edu. You may also send me email to ask questions.

1. Fill in the Blanks

(a) The items of a priority queue represent _____ .

2. In the array implementation of the binary tree implementation of a minheap, the number of items in the heap is 10 and the array contains the following characters in this order:

E J F Q K M N R V L

(a) Sketch the binary tree.

(b) Show the steps of inserting the symbol *H* into the minheap. Show both the array and the tree at each step.

(c) Starting with the heap after you inserted *H*, show the steps of *deletemin*.

3. A loop invariant has to be Boolean, that means that it's either true or false. But it doesn't have to be an equation – it could be simply a sentence. Here is an example.

```
int A[N]; // assume that the values of this array have been given
```

```
int sumA()
{
    int i = 0;
    int sum = 0;
    // Loop invariant holds here
    while (i < N)
    {
        // start of loop
        sum = sum + A[i];
        i++;
        // end of loop
    }
    // Loop invariant holds here
    return sum;
}
```

The loop invariant is the following statement: “sum is the sum of the first i values of the array A”

If the invariant holds at the start of any iteration of the loop, it holds at the end of that iteration.

Here is another code fragment. The code is for selection sort. and its purpose is to sort the array A in increasing order. There are two loops, and each has a loop invariant. Find those two loop invariants.

```
void selectionsort() // sorts the array A[N]
{
    int i = 0;
    while(i < N)
    {
        int j = i+1;
        while(j < N)
        {
            if(A[j] < A[i]) swap(A[j],A[i]);
            j++;
        }
        i++;
    }
}
```