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

Assignment 7: Due Saturday April 29, 2023, 11:59 PM

More problems may be added later.

Name:____

You are permitted to work in groups, get help from others, read books, and use the internet.

1. Fill in the blanks. For the first three of these questions, give the exact formula.

   (a) A binary tree of height h has at most ______________________ nodes.

   (b) A graph (not ordered graph) with n nodes has at most ______________________ edges.

   (c) A planar graph with 2 nodes has at most 1 edge, while a planar grap with 3 nodes has at most 3 edges. If $n \geq 3$, a planar graph with n nodes has at most ______________________ edges.

   For these questions, assume the graph has n vertices and m edges.

   (d) The asymptotic complexity of the Floyd Warshall algorithm is ______________________.

   (e) The asymptotic complexity of the Bellman Ford algorithm is ______________________.

   (f) The asymptotic complexity of Dijkstra's algorithm is ______________________.

   (g) The asymptotic complexity of Johnson's algorithm is ______________________.

   For these questions, the number of items is n. Give the worst case complexity for each.

   (h) The asymptotic complexity of bubblesort is ____________.

   (i) The asymptotic complexity of selection sort is ____________.

   (j) The asymptotic complexity of tree sort is ____________.

   (k) The asymptotic complexity of merge sort is ____________.

   (l) The asymptotic complexity of quicksort is ____________.

   (m) The asymptotic complexity of polyphase merge sort is ____________.

   (n) The asymptotic complexity of the BFPRT algorithm is ____________

2. Give the asymptotic complexity, in terms of `n`, of each of the following code fragments.

   (a)
   ```
   int kount = 0;
   for(int i = 2; i < n; i = i*i)
     kount++;
   cout << kount;
   ```

   (b)
   ```
   int kount = 0;
   for(int i = 1; i < n; i++)
     for(int j = 1; j < i; j = 2*j)
       kount++;
   cout << kount;
   ```

(c)

```
int kount = 0;
for(int i = 1; i < n; i++)
  for(int j = i; j < n; j = 2*j)
    kount++;
cout << kount;
```

(d)

```
int kount = 0;
for(int i = 1; i*i < n; i++)
  kount++;
```

(e)

```
int kount = 0;
for(int i = 1; i < n; i++)
  for(int j = n; j > i; j = j/2)
kount++;
```

(f)

```
int kount = 0;
 for(int i = 1; i < n; i++)
   for(int j = i; j > 0; j = j/2)
kount++;
```

3. What properties are desirable for a hash function h for a hash table used as a search structure?

4. The following code could be used as a subroutine for both quicksort and select. Assume A[n] is an array of integers. For simplicity, we assume that no two entries of A are equal. Write a loop invariant for the while loop.

```
int pivot = A[0];
int lo = 0;
int hi = n-1;
while(lo < hi)
 {
  if(A[lo+1] < pivot) lo++;
  else if(A[hi] > pivot) hi--;
  else swap(A[lo+1],A[hi]);
 }
```

5. The main memory of your computer is probably a 1-dimensional array. That is, each variable of your program is stored in a location `RAM[i]` for some $0 \leq i < N$, where N is the size of your memory.

   Suppose you declare an array

   `int A[20][100][40];`

   Assume indices start at zero as in C++.

   (a) If the compiler decides to store A in 80,000 consecutive locations, starting at `RAM[4000]`, in **row-major** order, where would `A[13][45][22]` be stored?

   (b) On the other hand, suppose the compiler decides to store A in **column-major** order starting at `RAM[4000]`. In that case, where would `A[13][46][22]` be stored?

6. Explain how to use a search structure to implement a sparse array.

7. Draw figures illustrating insertion (enqueue) into a queue implemented as singly linked circular list with dummy node.

   Start with a figure illustrating the structure when the items, from front to rear, are B, M, Q, R.

   Next, show the steps needed to insert H.