CSCI 470 Practice Exam 02

Date: October 27, 2023

Full Name:

Student ID:

DIRECTIONS:

- Write your full name, and student ID above.
- Write your answers on the exam paper.
- You are allowed one cheat sheet.
- If you need extra space, please use the back of a page.
- You have 80 minutes to complete the exam.
- Please do not turn the exam over until you are instructed to do so.
- Good Luck!

1. For each of the following problems, answer **True** or **False** and BRIEFLY JUSTIFY your answer.

   (a) The smallest possible depth of a leaf in a decision tree for a comparison sort is $n - 1$, where n is the number of elements in an array to sorted.

   (b) A binary-search tree will always have its height $O(\lg n)$.

   (c) Searching an element in a linked-list with n objects takes, $\Theta(n \lg n)$.

(d) The total running time of the BFS procedure is $O(V + E)$.

(e) The following procedure returns the successor of node x in a binary-search tree.

```
TREE-SUCCESSOR(x)
1  if x.right ≠ NIL
2      return TREE-MAXIMUM(x.right)
3  y = x.p
4  while y ≠ NIL and x == y.right
5      x = y
6      y = y.p
7  return y
```

2. Here is the pseudocode of COUNTING-SORT:

```
COUNTING-SORT(A, B, k)
 1  let C[0..k] be a new array
 2  for i = 0 to k
 3      C[i] = 0
 4  for j = 1 to A.length
 5      C[A[j]] = C[A[j]] + 1
 6  // C[i] now contains the number of elements equal to i
 7  for i = 1 to k
 8      C[i] = C[i] + C[i − 1]
 9  // C[i] now contains the number of elements less than or equal to i
10  for j = A.length downto 1
11      B[C[A[j]]] = A[j]
12      C[A[j]] = C[A[j]] − 1
```

Suppose that we were to rewrite the **for** loop header in line 10 of the COUNTING-SORT as

```
10  for j = 1 to A.length
```

Show that the algorithm still works properly.

(Please use this page if the previous page is not enough to write your answers.)

3. Hash tables

   (a) Briefly describe what is a collision in a hash table.

   (b) Argue why collision is inevitable even with a perfect hashing.

(c) Describe a method to resolve collision in a hash table.

(d) State the runtime of a successful search in a hash table in terms of the ***load factor***, α. If $n = O(m^3)$, where n is the number of keys, and m is the number of slots in a hash table, what would be the upper bound of searching an element in the hash table on average?

4. BFS

   (a) Write the pseudocode to find the shortest path from s to v, given we have already run a breadth-first search on the graph.

(b) Find out if there is a shortest path from node P to U using a BFS procedure. You must use a predecessor graph to figure out if there is a shortest path from P to U or not.

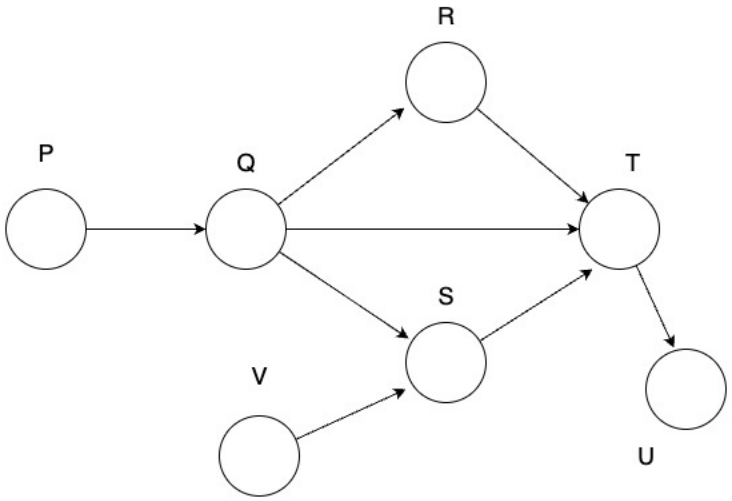

Figure 1

(Please use this page if the previous page is not enough to write your answers.)

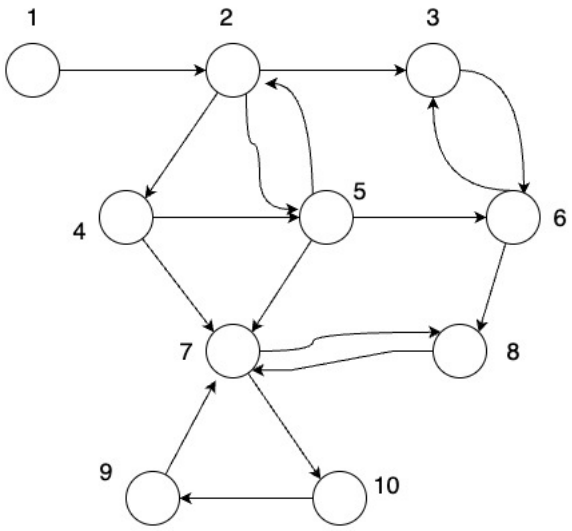

Figure 2

5. DFS

   (a) Run a DFS on the graph above. Assume that the iteration over the nodes is in an increasing order, and assume that each adjacency list is also ordered in an increasing order.

(b) Transpose the graph above.

(c) Run another DFS on the graph based on the decreasing finishing times of the nodes computed in 5(a).

(d) List the nodes of each tree in the depth-first search forest created by the DFS on G^T.

(e) Write the ordering of vertices which are topologically sorted using the computation in 5(a), if there is a valid topological order.

(Please use this page if the previous page is not enough to write your answers.)