Practice Final Exam: Part I and II

Date: November 29, 2023

DIRECTIONS:

- Write your answers on the exam paper.
- If you need extra space, please use the back of a page.
- You are allowed one cheat sheet.
- 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. MAXIMUM(A) procedure iterates over an array to find the largest integer in the array.

   Input: A is a input array with indices $S[6, 4, 1, 2, 10, 3, 5]$
   Output: 10

   MAXIMUM(A)

   ```
   1  maxval = A[1]
   2  for i = 2 to A.length
   3       if A[i] ≥ maxval
   4            maxval = A[i]
   5  return maxval
   ```

   Here is a loop invariant for MAXIMUM procedure:

   > Before j-th iteration, $maxval$ is greater than or equal to all elements in the subarray $A[1, . . , j - 1]$.

   Use the loop invariant to prove that your algorithm is correct. Make sure that your loop invariant fulfills the three necessary properties.

(Please use this page to write your answer for question (2) if the previous page is not enough.)

2. Use either substitution method or a recursion tree to show that $T(n) = 2T(n/2) + n^2$ is $T(n) = O(n^2)$.

3. Priority Queues

   (a) For a priority queue data structure, write pseudocode for HEAP-EXTRACT-MAX(S), where S is a set of elements, each with an associated value called a ***key***.

   (b) State an upper bound for the runtime of HEAP-EXTRACT-MAX(S) procedure. Justify your answer.

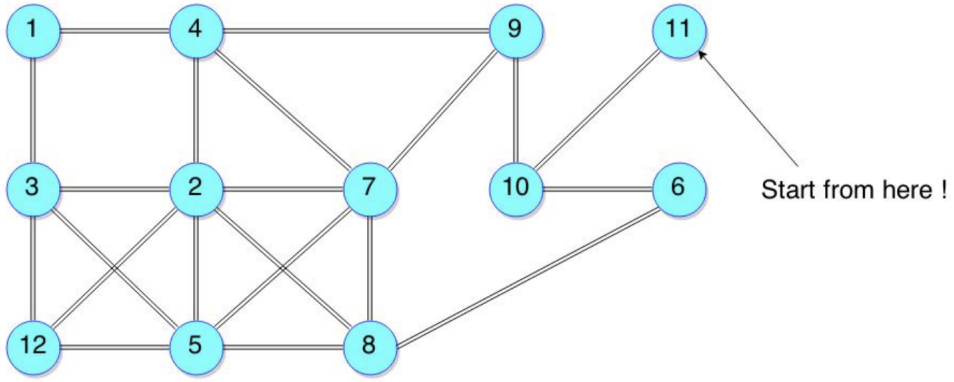
4. What is the running time of QUICKSORT when all elements of array A have the same value?

5. Write a procedure called LIST-DELETE removes an element x from a linked list L. A pointer to x is given.

6. Write a procedure to find the predecessor of a key, x in a binary-search tree. State the upper bound of the runtime to this procedure.

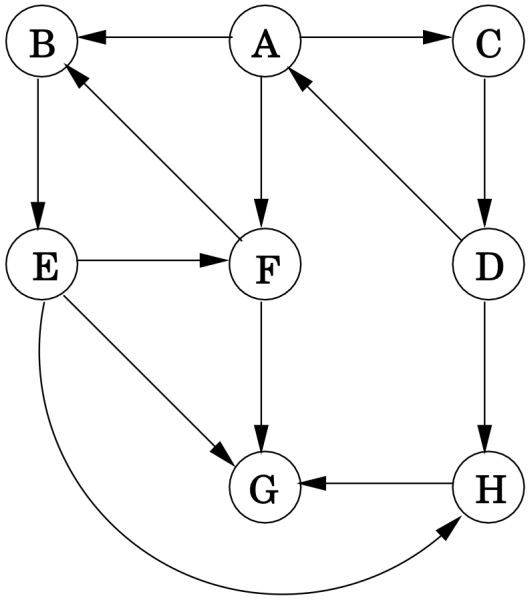
7. A Catalan number is recursively defined as follows:
   $C_0 = 1$, $C_1 = 1$, and $C_n = \sum_{i=0}^{n-1} C_i * C_{n-i-1}$.
   Provide a $O(n)$ dynamic-programming procedure to calculate the n-th Catalan number.

8. Run BFS (Breadth-first search) procedure on the following undirected graph and write out the $v.d$ and $v.\pi$ for each vertex in the graph. Here $v.d$ is the **distance** from the source node, and $v.\pi$ is the predecessor of a node. The source node is 11 as stated in graph below.



9. DFS

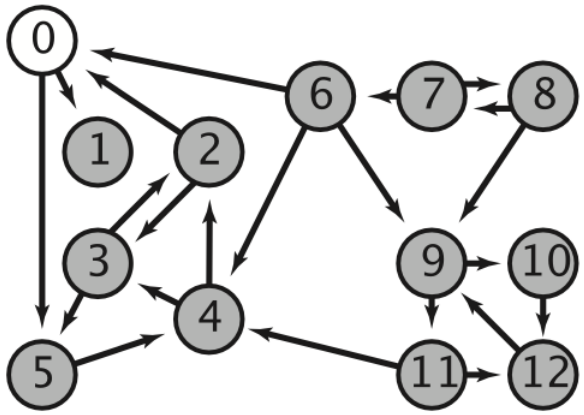
   (a) Run DFS (Depth-first search) procedure on the following directed graph and write out the $v.d$ and $v.f$ for each vertex in the graph. Here, $v.d$ is the discovery time, and $v.f$ is the finishing time. The source node here is node A.



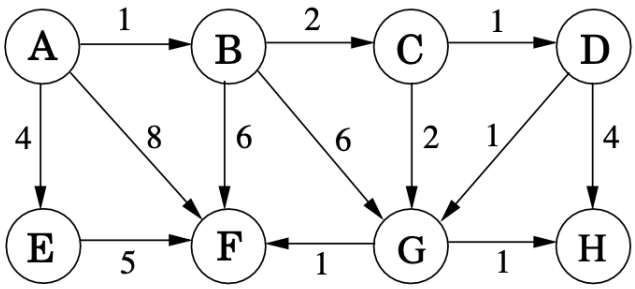
   (b) Write out the topological ordering of the nodes based on the DFS run in the previous question.

   Note: There is no topological ordering for this graph, because it's not a directed acyclic graph (DAG). If there was no edge going from D to A, then, a topological ordering is possible.

10. Find the strongly connected components in the following graph using the STRONGLY-CONNECTED-COMPONENTS procedure.



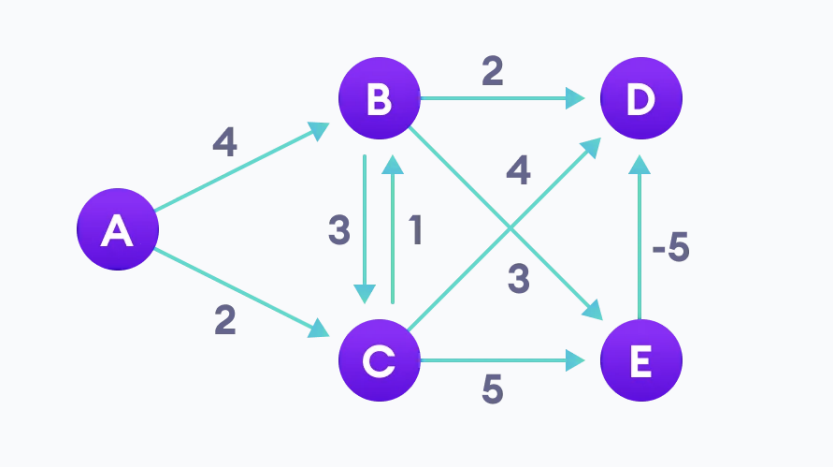
11. Run DIJKSTRA procedure in the following graph, with the source node A:



A	B	C	D	E	F	G	H
0	∞	∞	∞	∞	∞	∞	∞
0							
0							
0							
0							
0							
0							
0							
0							

A	B	C	D	E	F	G	H
NIL	NIL	NIL	NIL	NIL	NIL	NIL	NIL
NIL							
NIL							
NIL							
NIL							
NIL							
NIL							
NIL							
NIL							

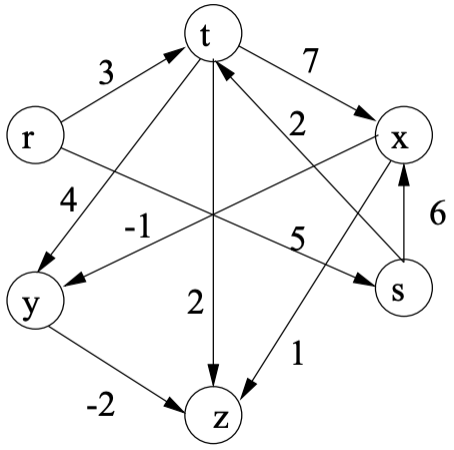
12. Run BELLMAN-FORD procedure in the following graph, with the source node A:



A	B	C	D	E
0	∞	∞	∞	∞
0				
0				
0				
0				

A	B	C	D	E
NIL	NIL	NIL	NIL	NIL
NIL				
NIL				
NIL				
NIL				

13. Run DAG-SHORTEST-PATHS procedure in the following graph, with the source node r



14. Run a bottom-up LCS-LENGTH procedure on the following two sequences to find the longest common subsequence using a table.

    $Y = CGATAATTGAGA$
    $X = GTCCTAATA$