BITS Pilani, K. K. Birla Goa Campus

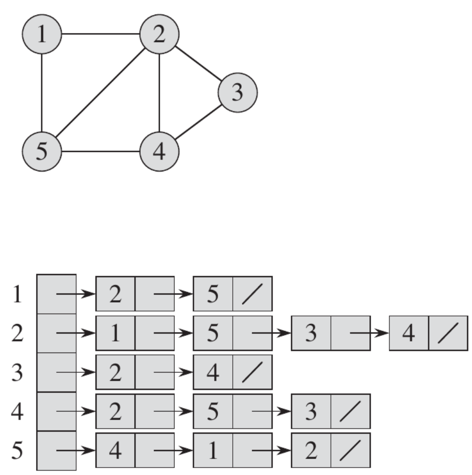
Data Structures and Algorithms (CS F211)

Midsem Exam (16/03/2023)
Total Marks: 27
Time Limit: 90 minutes

This question paper contains **3** questions carrying **27** marks. All questions are compulsory. Answer each question on a fresh page. Answer all the parts of a question in the same place. Show the necessary steps and justifications for your answers.

Question 1 (9 marks)

Consider the graph $G(V, E)$ and the corresponding adjacency-list representation shown below. The figure below also shows the (possibly incorrect) BFS procedure that uses just *two* colors to perform the Breadth-first search.



```
BFS(G, s)
for each vertex u ∈ G.V − {s}
    u.color = WHITE
    u.d = ∞
    u.π = NIL
s.color = WHITE
s.d = 0
s.π = NIL
Q = ∅
ENQUEUE(Q, s)
while Q ≠ ∅
    u = DEQUEUE(Q)
    for each v ∈ G.Adj[u]
        if v.color == WHITE
            v.d = u.d + 1
            v.π = u
            ENQUEUE(Q, v)
    u.color = BLACK
```

(a) (2 marks) Let vertex 2 be the source vertex. What will be the value of the distance attribute ($v.d$) for vertices 4 and 5 when the BFS procedure shown above terminates. (Assume that the algorithm uses the adjacency-list representation shown above.)

(b) (2 marks) In part (a), what will be the value of the parent attribute ($v.\pi$) for vertices 4 and 5 when the BFS procedure shown above terminates.

(c) (1 mark) In part (a), how many times will vertex 4 be enqueued before the given BFS procedure terminates.

(d) (2 marks) How should you modify the given BFS procedure so that the algorithm finds the shortest distance of each vertex from the source vertex. Make minimum modifications to the given BFS procedure. Mention the line number(s) and the modification(s) that must be made. Justify your suggested modification(s). (The modified BFS procedure should also use only *two* colors.)

(e) (2 marks) Is $\log_4 n = \Theta(\log_{16} n)$? (Use the set theoretic definition of the Θ-notation to justify your answer. Show the necessary steps.)

Question 2 (9 marks)

Suppose we want to devise an algorithm to build a *ternary* max-heap data structure using the elements of an *unsorted* input array $A[1..n]$. Each array element i can have three child nodes : left $(3i-1)$, mid $(3i)$ and right $(3i+1)$. The max-heap property $A[\text{PARENT}(i)] \geq A[i]$ must be satisfied by all the non-root nodes in the ternary max-heap.

Suppose we use the BUILD-MAX-HEAP procedure given below to build the *ternary* max-heap in an *efficient* manner.

function BUILD-MAX-HEAP(A)
  A.heap-size = *A.length*
  for $i =$ ______ **downto/to** ______ **do**
    MAX-HEAPIFY(A, i)

(a) (1 mark) Rewrite the BUILD-MAX-HEAP procedure by filling in the blanks such that a ternary max-heap can be constructed efficiently. Assume that the MAX-HEAPIFY procedure mentioned above is suitably modified to operate on a ternary max-heap.

(b) (4 marks) Show that there are at most $\left\lceil \dfrac{2n-1}{3^{h+1}} \right\rceil$ nodes of height h in any n-element ternary max-heap. (Write the steps clearly with proper justifications.)

(c) (4 marks) Find a tight upper bound for the running time of BUILD-MAX-HEAP procedure for a ternary max-heap. You can use the result in part (b) and the following result if required :

$$\sum_{k=0}^{\infty} kx^k = \frac{x}{(1-x)^2}$$

(Write the steps clearly with proper justifications.)

Question 3 (9 marks)

Write the pseudocode for FIND-MEDIAN procedure that finds the *median* element of an unsorted input array $A[1..n]$. Let the median element be defined as the $\left\lceil \dfrac{n}{2} \right\rceil^{th}$ smallest element of the input array A. The FIND-MEDIAN procedure should follow a divide-and-conquer approach based on the PARTITION procedure. (Assume PARTITION procedure is same as that used in the QUICKSORT algorithm.)

(a) (4 marks) Write the pseudocode for the FIND-MEDIAN procedure. (The *average-case* running time of the FIND-MEDIAN procedure must be $O(n)$.)

(b) (2½ marks) Find the recurrence for the *average-case* running time of the FIND-MEDIAN procedure found in part (a). Solve the recurrence using the *recursion tree* method to find the average-case running time. Write the steps clearly with proper justifications.

(c) (2½ marks) Find the recurrence for the *worst-case* running time of the FIND-MEDIAN procedure found in part (a). Solve the recurrence using the *substitution method* to find the worst-case running time. Write the steps clearly with proper justifications.