CSE310 Midterm02A, 03/30/2021, 4:30pm, Due: 03/30/2021, 5:50pm

This is the question document. This is an open book exam. It should be your own work. Textbooks/notes/computers are allowed. **You have to use the companion answer sheet (which is a fillable PDF file) to check/enter your answers to the questions described here**. Adobe Acrobat Reader can be found at `https://get.adobe.com/reader/`. Hand-written answers (or photo/scan of it) will not be graded. **Submit your answer sheet on Gradescope, following the link on Canvas**. You should name your file using the format **CSE310-MDT02A-LastName-FirstName.pdf**. **The submission deadline is 5:50pm.**

Q1 (15 points) In class, we have studied max-heap and its operations in details. The min-heap data structure is defined similarly, `with max replaced by min, greater than replaced by less than, etc`. The operations of min-heap are also symmetric to the corresponding operations of max-heap. This question is about min-heap. A min-heap with `capacity` 20 and `size` 12 is shown in the following array.

i	1	2	3	4	5	6	7	8	9	10	11	12
$A[i]$	88	89	93	91	90	94	99	92	96	98	95	97

The following three sub-questions all refer to this min-heap (not the heap you obtained after doing some operations).

(a) On the answer sheet, show the result after applying `heap-extract-min(A)` to the min-heap at the start of this question.

(b) On the answer sheet, show the result after applying `heap-decrease-key(A, 10, 85)` to the min-heap at the start of this question.

(c) On the answer sheet, show the result after applying `min-heap-insert(A, 80)` to the min-heap at the start of this question.

Q2 (15 points) This question is about disjoint set operations. Assume that we are using **union by rank** and **find with path compression**. Suppose that you are given a disjoint set structure described by the following array. The following three sub-questions all refer to **this disjoint set** (not the disjoint set you obtained after doing some operations).

i	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
$A[i]$	-3	1	1	3	1	5	5	7	1	9	9	11	-2	13	13	15

(a) On the answer sheet, show the result after applying `union(8, 16)` to the disjoint set at the start of this question.

(b) On the answer sheet, show the result after applying `union(10, 14)` to the disjoint set at the start of this question.

(c) On the answer sheet, show the result after applying `find-set(8)` and `find-set(10)` to the disjoint set at the start of this question.

Q3 (18 points) A directed graph $G = (V, E)$ is shown in Figure 1. Its adjacency lists are given in alphabetical order.

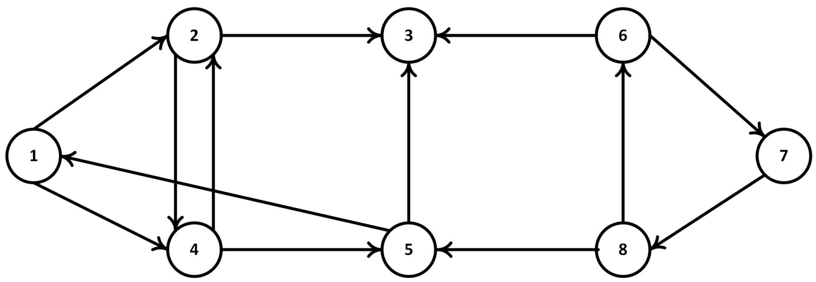

Figure 1: A directed graph

Apply depth-first search (DFS) on graph G. **In the main-loop of DFS, check the vertices in alphabetical order.**

During the computation, the algorithm computes values $v.dsc$, $v.fin$ and $v.\pi$ for vertices $v \in V$, where $v.dsc$ is the discovery time of vertex v, $v.fin$ is the finish time of vertex v, and $v.\pi$ is the predecessor of vertex v, respectively.

(a) On the answer sheet, enter the discovery times of selected vertices computed by the DFS.

(b) On the answer sheet, enter the finish times of selected vertices computed by the DFS.

(c) On the answer sheet, enter the predecessors of selected vertices computed by the DFS. If the value is `nil`, write `nil` (not `NIL`).

Q4 (12 points) An undirected graph $G = (V, E)$ is shown in Figure 2. Its adjacency lists are given in alphabetical order.

Apply breadth-first search (BFS) on graph G, starting from vertex `s=1`.

During the computation, the algorithm computes values $v.d$ and $v.\pi$ for vertices $v \in V$, where $v.d$ is the current distance from `s` to vertex v and $v.\pi$ is the current predecessor of vertex v.

(a) On the answer sheet, answer questions regarding the distance attributes of selected vertices computed by the BFS. If the value is ∞, write `infinity`.

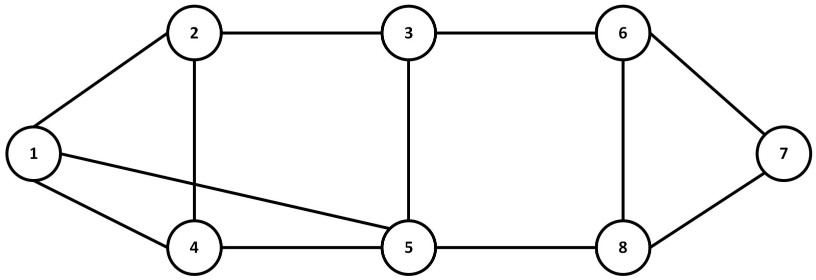

Figure 2: An undirected graph

(b) On the answer sheet, answer questions regarding the predecessor attributes of selected vertices computed by the BFS. If the value is `nil`, write `nil` (do not write `NIL`, `null`, or `NULL`).

Q5 (20 points) An edge weighted directed graph $G = (V, E, w)$ is shown in Figure 3. Its adjacency lists are given in alphabetical order.

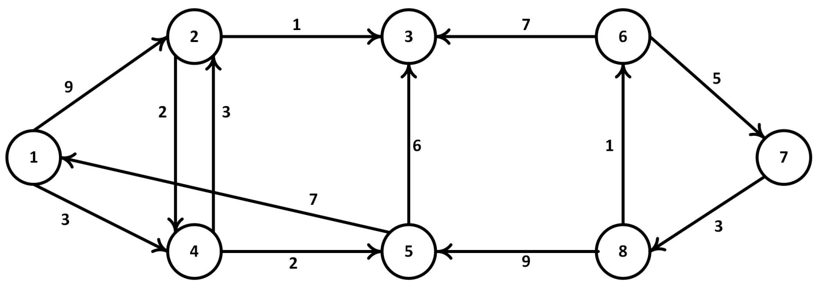

Figure 3: An edge weighted directed graph

Assume that you are using Dijkstra's algorithm to compute single-source shortest paths from source node `s=1`. During the computation, the algorithm computes values $v.d$ and $v.\pi$ for vertices $v \in V$, where $v.d$ is the current distance from `s` to vertex v and $v.\pi$ is the current predecessor of vertex v.

On the answer sheet, answer questions regarding the values of $v.d$ and $v.\pi$ at specified stages of the algorithm, for specified vertices $v \in V$. If the value is ∞, write `infinity`. If the value is `nil`, write `nil` (do not write `NIL`, `null`, or `NULL`).

Q6 (20 points) An edge weighted undirected graph $G = (V, E, w)$ is shown in Figure 4. Its adjacency lists are given in alphabetical order.

Assume that you are using Prim's algorithm to compute a minimum spanning tree of graph G, starting with root node `r=1`. During the computation, the algorithm computes values $v.key$ and $v.\pi$ for vertices $v \in V$, where $v.key$ is the distance from vertex v to the current tree, and $v.\pi$ is the current predecessor of vertex v.

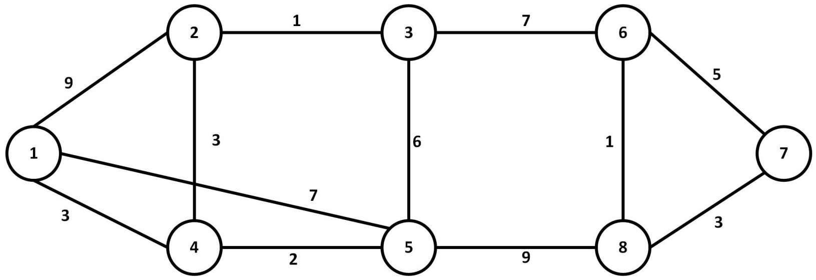

Figure 4: An edge weighted undirected graph

On the answer sheet, answer questions regarding the values of $v.key$ and $v.\pi$ at specified stages of the algorithm, for specified vertices $v \in V$. If the value is ∞, write `infinity`. If the value is `nil`, write `nil` (do not write `NIL`, `null`, or `NULL`).