Quiz 4 — Data Struct. & More (T. I/21–22)

Directions:

- This exam is "paper-based." Answer all the questions in the on-screen editor provided or pasting pictures/figures into the document.
- No consultation with other people is permitted. But feel free to use your notes, books, and the Internet. You are also allowed to write code and run it.
- You can chat with the instructors via the built-in chat.
- This quiz is worth a total of 35 points, but we'll grade out of 30. Anything above 30 is extra credit. You have 80 minutes. Good luck!

Problem 1: Graph Representation Quickies (3 points = 0.5 points/blank)

We have seen a number of graph representations. Consider the following two variants carefully:

(1) a *variant* **adjacency map** represents a graph as a `TreeMap` mapping each vertex to a `HashSet` of its neighbors

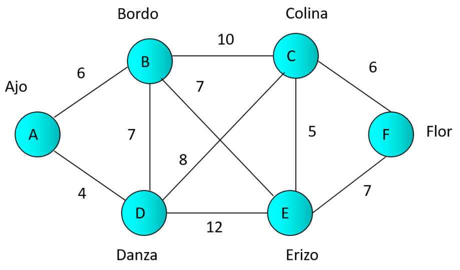
(2) an **edge list** represents a graph as an `ArrayList` of edges (an edge is a `Pair`)

Let $G = (V, E)$. You will complete the table below with the running time of **(i)** `G.deg(u)` for computing the degree of a vertex u in the graph G, **(ii)** `G.isEdge(u, v)` for determining whether there is an edge from u to v in G, and **(iii)** `G.allEdges()` for returning a brand new list (`ArrayList`) of all edges of G. By "brand new", we mean this list is (re-)created fresh when the method is called. Remember $n = |V|$ and $m = |E|$.

Operation	Adjacency Table	Edge List
`deg(u)`	$O(\_\_\_\_\_\_)$	$O(\_\_\_\_\_\_)$
`isEdge(u, v)`	$O(\_\_\_\_\_\_)$	$O(\_\_\_\_\_\_)$
`allEdges()`	$O(\_\_\_\_\_\_)$	$O(\_\_\_\_\_\_)$

Problem 2: Minimum Spanning Tree (7 points)

You are about to install cable TV lines connecting six towns. Each edge represents the cost of installing a cable between a pair of towns. How can we pick a route that minimizes the cost of installing the cable system? Indicate which edges are in your MST and the total weight of your tree. Use the following format to list the edges in the MST: (example) `A=D`, `D=B`, `B=C`, `C=E`, `E=F`, total weight = 33. Here, the `A=D` means installing a cable between town `A` and town `B`.



Quiz 4 — Data Struct. & More (T. I/21–22)

Directions:

- This exam is "paper-based." Answer all the questions in the on-screen editor provided or pasting pictures/figures into the document.
- No consultation with other people is permitted. But feel free to use your notes, books, and the Internet. You are also allowed to write code and run it.
- You can chat with the instructors via the built-in chat.
- This quiz is worth a total of 35 points, but we'll grade out of 30. Anything above 30 is extra credit. You have 80 minutes. Good luck!

Problem 1: Graph Representation Quickies (3 points = 0.5 points/blank)

We have seen a number of graph representations. Consider the following two variants carefully:

(1) a *variant* **adjacency map** represents a graph as a `TreeMap` mapping each vertex to a `HashSet` of its neighbors

#edges

(2) an **edge list** represents a graph as an `ArrayList` of edges (an edge is a `Pair`)

vertices set of directed edges

Let $G = (V, E)$. You will complete the table below with the running time of **(i)** `G.deg(u)` for computing the degree of a vertex u in the graph G, **(ii)** `G.isEdge(u, v)` for determining whether there is an edge from u to v in G, and **(iii)** `G.allEdges()` for returning a brand new list (`ArrayList`) of all edges of G. By "brand new", we mean this list is (re-)created fresh when the method is called. Remember $n = |V|$ and $m = |E|$.

Hash set → sorted

→Tree map

Array List

{ (v,v), (v,v), (v,v) }

Operation	Adjacency Table	Edge List
G.get(u) . size() `deg(u)`	$O(\underline{\log n})$	$O(\underline{m})$
G.get(u). contain(v) `isEdge(u, v)`	$O(\underline{\log n})$	$O(\underline{m})$
`allEdges()`	$O(\underline{m \log n})$	$O(\underline{m})$

contain O(1)

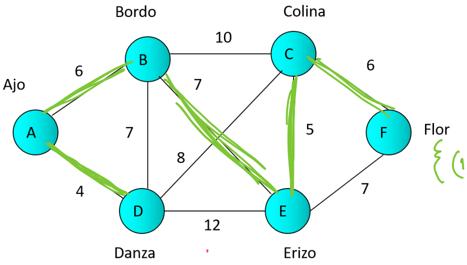
TreeMap

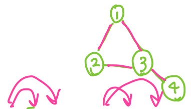
contain keys
put
get
remove } log(n)

{ v:{·~, ~, ~}, v: {~, ~, ~}}

Problem 2: Minimum Spanning Tree (7 points)

You are about to install cable TV lines connecting six towns. Each edge represents the cost of installing a cable between a pair of towns. How can we pick a route that minimizes the cost of installing the cable system? Indicate which edges are in your MST and the total weight of your tree. Use the following format to list the edges in the MST: (example) A=D, D=B, B=C, C=E, E=F, total weight = 33. Here, the A=D means installing a cable between town A and town B.





4

1=[2,3] , 2=[3,4]

{ (1,2), (1,3), (2,3), (2,4) }

n

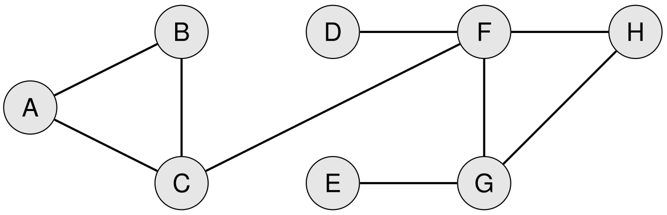
Problem 3: Representation & Breadth-First Search (7 points)

(a) **[2 points]** Given the following adjacency matrix of four vertices A, B, C and D, what is the corresponding adjacency array?

$$\begin{array}{c|cccc} & 0 & 1 & 2 & 3 \\ \hline A & 0 & 1 & 0 & 0 \\ B & 1 & 0 & 1 & 1 \\ C & 0 & 1 & 0 & 0 \\ D & 0 & 1 & 0 & 0 \end{array} \Longrightarrow$$

Vertex	**List of adjacent vertices**	
A	B	1
B	A C D	0 2 3
C		1
D		1

(b) **[5 points]** On the graph below, suppose BFS is started from vertex D, so frontier $F_0 = \{\text{D}\}$. Indicate all the frontiers after that, until the BFS algorithm terminates.



F_0	{D}
F_1	{F}
F_2	{H, C, G}
F_3	{A, B, E}

Problem 4: Max-Heap Priority Queue (9 points)

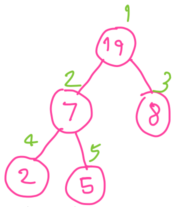
The max-heap priority queue data structure, as discussed in class, maintains a binary heap tree whose root stores the largest value.

(a) **[3 points]** Suppose our max-heap priority queue stores the following values: 2, 19, 7, 5, 8. Draw one *valid* heap tree corresponding to this max-heap priority queue.

(b) **[3 points]** A heap tree is often stored in a flat array such as an `ArrayList` with index 0 left empty (a sentinel), so the root is kept at index 1. Shown below, `heapArray` is an `ArrayList` storing your tree from the previous part. Fill out the values of `heapArray`.

`heapArray` =

Index	0	1	2	3	4	5
Value	×	19	7	8	2	5



(c) **[3 points]** **Claim:** In a (max) heap tree, the smallest value (for example, the number 2 in the above set of values) is at the bottom-left node.

Prove this claim mathematically. Or refute it by providing a counterexample.

every parent's value is greater or equal to its child's value

Heap may or may not be a sorted array

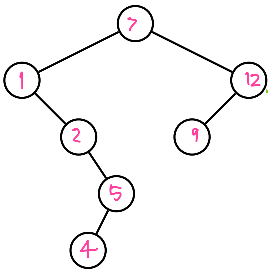
Problem 5: Binary Search Trees (9 points)

(a) **[3 points]** Consider the binary tree structure below. Assign the following keys

$$7, 2, 5, 9, 12, 1, 4$$

1 2 4 5 7 9 12

to each node so that the tree is a binary *search* tree.



(b) **[3 points]** Now that you have a BST, write down the keys visited in a pre-order traversal.

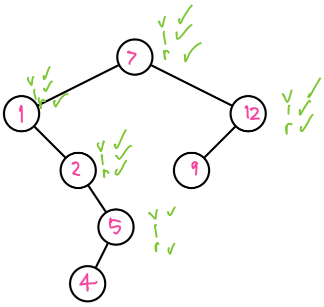
visit
left
right

(c) **[3 points]** For the same BST, write down the keys visited in a post-order traversal.

left
right
visit

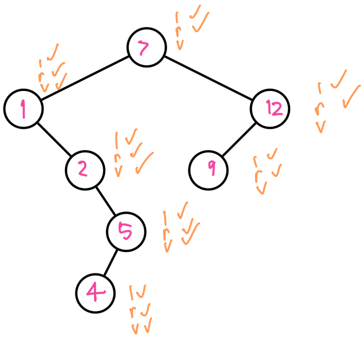
pre-order

b) 7 1 2 5 4 12 9



post-order

c) 4 5 2 1 9 12 7


7
1
12
2
9
5
4
l
r
v