

1 Exercises

Exercise 1. Consider the following undirected graph:

```
13 13
0 1
0 2
0 5
0 6
4 3
5 3
5 4
6 4
7 8
9 10
9 11
9 12
11 12
```

- What is the adjacency matrix representation of the graph?
- What is the adjacency list representation of the graph? List the adjacent vertices in decreasing order of their IDs.
- What is the state of the `edgeTo` array if you run depth-first search (DFS) on the graph, starting at vertex 0? Is there a path from 0 to 3? If so, what is it?
- What is the state of the `distTo` and `edgeTo` arrays if you run breadth-first search (BFS) on the graph, starting at vertex 0? Is there a path from 0 to 3? If so, what is it?
- How many connected components does the graph have?
- What is the component identified by a DFS on the graph, starting at vertex 3?

Exercise 2. Consider building a `SymbolGraph` object `sg` from the following representation of a symbol graph as a file:

```
JFK MCO
ORD DEN
ORD HOU
DFW PHX
JFK ATL
ORD DFW
ORD PHX
ATL HOU
DEN PHX
PHX LAX
JFK ORD
DEN LAS
DFW HOU
ORD ATL
LAS LAX
ATL MCO
HOU MCO
LAS PHX
```

- What is the state of the `sg.st` symbol table and `sg.keys` array?
- Who are the neighbors of `HOU`?

2 Solutions to Exercises

Solution 1.

a.

	0	1	2	3	4	5	6	7	8	9	10	11	12
0	0	1	1	0	0	1	1	0	0	0	0	0	0
1	1	0	0	0	0	0	0	0	0	0	0	0	0
2	1	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	1	1	0	0	0	0	0	0	0
4	0	0	0	1	0	1	1	0	0	0	0	0	0
5	1	0	0	1	1	0	0	0	0	0	0	0	0
6	1	0	0	0	1	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	1	0	0	0	0
8	0	0	0	0	0	0	0	1	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	1	1	1	1
10	0	0	0	0	0	0	0	0	0	1	0	0	0
11	0	0	0	0	0	0	0	0	0	1	0	0	1
12	0	0	0	0	0	0	0	0	0	1	0	1	0

b.

```

0: 6 5 2 1
1: 0
2: 0
3: 5 4
4: 6 5 3
5: 4 3 0
6: 4 0
7: 8
8: 7
9: 12 11 10
10: 9
11: 12 9
12: 11 9

```

c.

v	edgeTo[v]
0	0
1	0
2	0
3	5
4	6
5	4
6	0
7	0
8	0
9	0
10	0
11	0
12	0

Yes, 0 -> 6 -> 4 -> 5 -> 3

d.

v	distTo[v]	edgeTo[v]
0	0	0
1	1	0
2	1	0
3	2	5
4	2	6
5	1	0
6	1	0
7	∞	0
9	∞	0
10	∞	0
11	∞	0
12	∞	0

Yes, 0 -> 5 -> 3

e. 3

f. {0, 1, 2, 3, 4, 5, 6}

Solution 2.

a. sg.st:

```

ATL -> 7
DEN -> 3
DFW -> 5
HOU -> 4
JFK -> 0
LAS -> 9
LAX -> 8
MCO -> 1
ORD -> 2
PHX -> 6

```

sg.keys:

```

0: JFK
1: MCO
2: ORD
3: DEN
4: HOU
5: DFW
6: PHX
7: ATL
8: LAX
9: LAS

```

b. MCO, DFW, ATL, and ORD