Review of hashing, balancing, graphs

1. The Final Exam
   - Monday 11 December, BSB 337, from 8AM to 10AM

2. Examples of Questions
   - frequency tables and hash functions
   - balancing binary search trees
   - graph traversals and greedy algorithms

MCS 360 Lecture 43
Introduction to Data Structures
Jan Verschelde, 8 December 2017

Review of hashing, balancing, graphs

1. The Final Exam
   - Monday 11 December, BSB 337, from 8AM to 10AM

2. Examples of Questions
   - frequency tables and hash functions
   - balancing binary search trees
   - graph traversals and greedy algorithms

The Final Exam

- Monday 11 December, BSB 337, from 8AM to 10AM.
- The exam is closed book. No calculators, no computers.
- This review has its focus on hashing, balancing, and graphs.
  1. Lectures 27 to 29, 33 to 35, 38 to 40 correspond respectively to chapters 9, 11, and 12 in the textbook.
  2. Hashing: open addressing and chaining.
  3. Self balancing search trees: AVL, red-black trees, 2-3-4 trees.
  4. Graph representations, traversing graphs, and shortest paths and minimum spanning trees.
- Questions on this review are representative, but the list is by no means exhaustive.
- Review the quizzes, homework problems, and midterm exams.

Review of hashing, balancing, graphs

1. The Final Exam
   - Monday 11 December, BSB 337, from 8AM to 10AM

2. Examples of Questions
   - frequency tables and hash functions
   - balancing binary search trees
   - graph traversals and greedy algorithms

making a frequency table

1. Give C++ code to read in the name of a text file and to build a frequency table of all vowels (letters `a`, `e`, `i`, `o`, and `u`) that occur in the file.

open addressing

2. Consider a hash table of size 5.
   Place the keys 21, 22, and 1 in this table using open addressing.
   Draw the evolution of the state of the hash table
   as you insert 21, 22, and 1.

open addressing

2. Consider a hash table of size 5.
   Place the keys 21, 22, and 1 in this table using open addressing.
   Draw the evolution of the state of the hash table
   as you insert 21, 22, and 1.

		21 mod 5 = 1	22 mod 5 = 2	1 mod 5 = 1
0				
1		21	21	21
2			22	22
3				1
4				

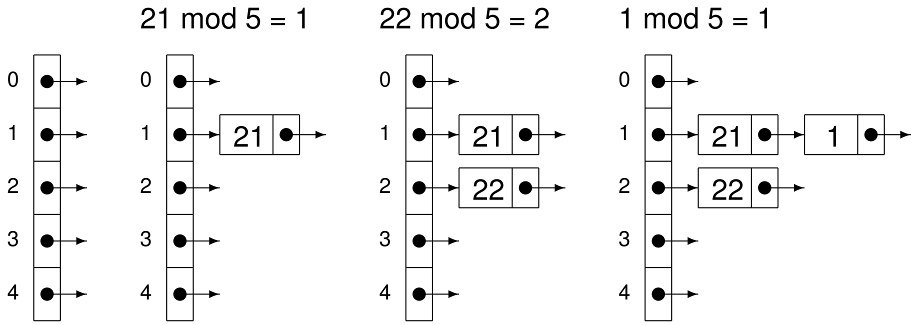
The number 1 is placed at the first open spot.

chaining

3. Consider a hash table of size 5.
   Place the keys 21, 22, and 1 in this table using chaining.
   Draw the evolution of the state of the hash table
   as you insert 21, 22, and 1.

chaining

3. Consider a hash table of size 5.
   Place the keys 21, 22, and 1 in this table using chaining.
   Draw the evolution of the state of the hash table
   as you insert 21, 22, and 1.



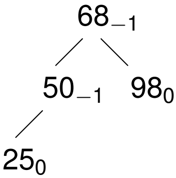
Review of hashing, balancing, graphs

1. The Final Exam
   - Monday 11 December, BSB 337, from 8AM to 10AM

2. Examples of Questions
   - frequency tables and hash functions
   - balancing binary search trees
   - graph traversals and greedy algorithms

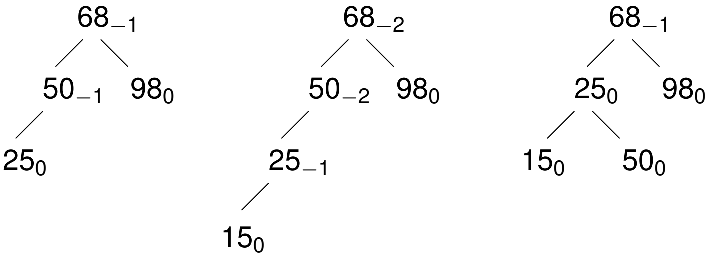
inserting into an AVL tree

4. Insert 15 into the AVL tree drawn below:



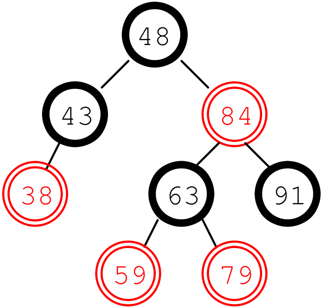
inserting into an AVL tree

4. Insert 15 into the AVL tree drawn at the left:



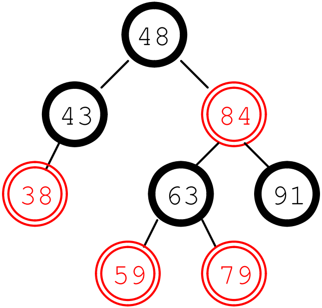
red-black trees

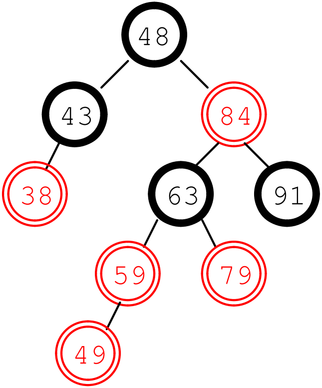
5. Consider the red-black tree (red nodes have hollow rings):



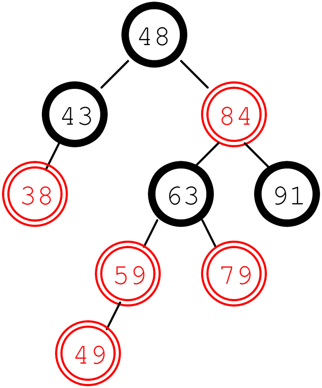
   1. Insert 49 in the tree. Draw all intermediate stages.
   2. Draw the 2-3-4 tree equivalent to the red-black tree above.

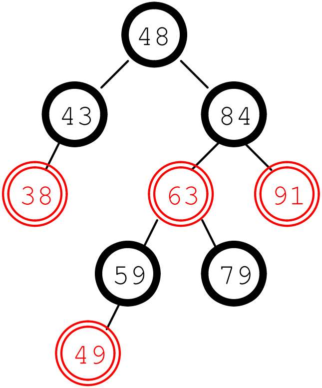
insert 49 in a red-black tree



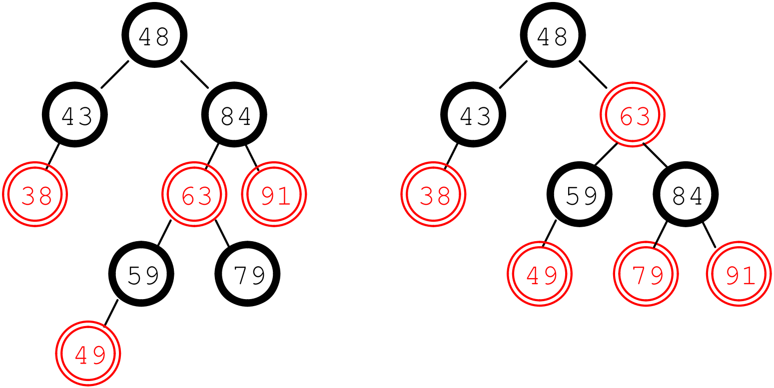


changing colors





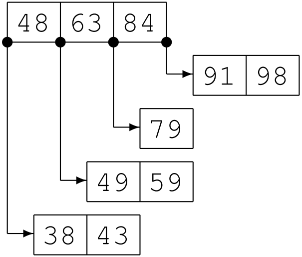
the right tree is left heavy: rotate



Observe that the color of 79 changed.

2-3-4 trees

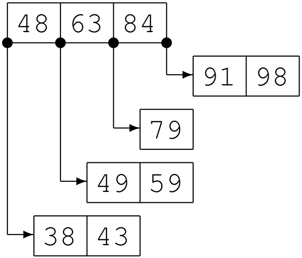
6. Insert the numbers 50, 51, 52, 53, 54, and 55 in the 2-3-4 tree drawn below:

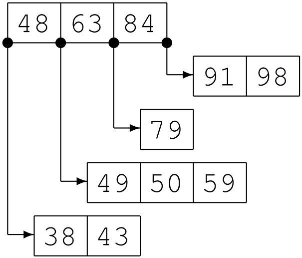


inserting into a 2-3-4 tree

insert the numbers 50, 51, 52, 53, 54, and 55

Inserting 50:

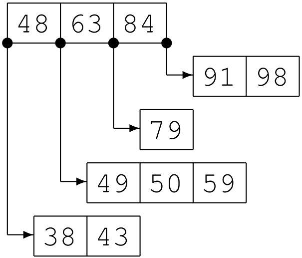


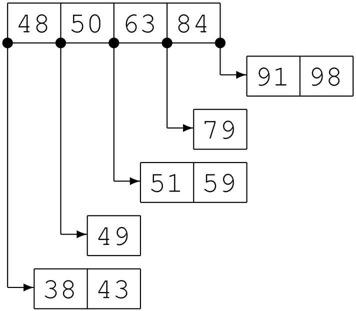


inserting into a 2-3-4 tree

insert the numbers 51, 52, 53, 54, and 55

Inserting 51:

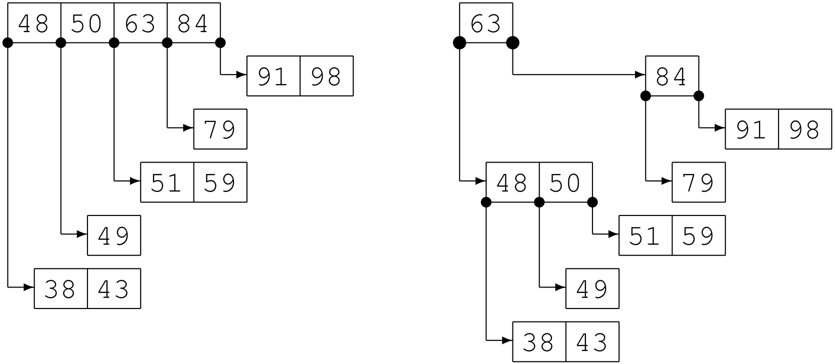




The root is a 5-node, we need to split the root.

splitting the root

The new root is 63:



Review of hashing, balancing, graphs

1. The Final Exam
   - Monday 11 December, BSB 337, from 8AM to 10AM

2. Examples of Questions
   - frequency tables and hash functions
   - balancing binary search trees
   - graph traversals and greedy algorithms

adjacency matrices

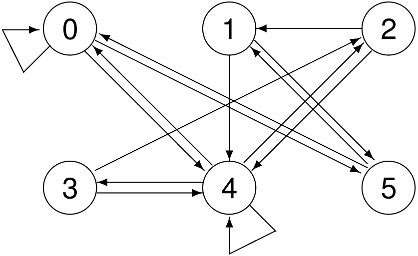
7. Consider the matrix

$$A = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

Draw a graph which has A as adjacency matrix.

a graph for a given adjacency matrix

$$A = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$



breadth-first search

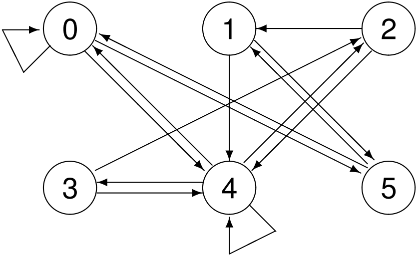
8. Consider the matrix

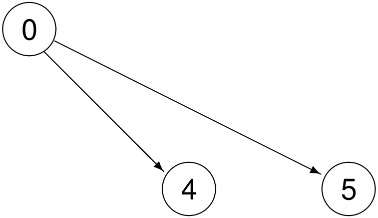
$$A = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

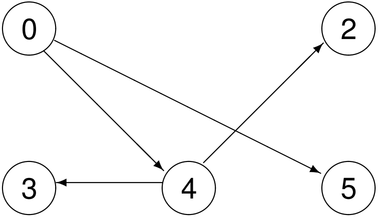
   Draw the breadth-first search tree for the graph, starting from vertex 0.

construction of the depth-first search tree

$$A = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

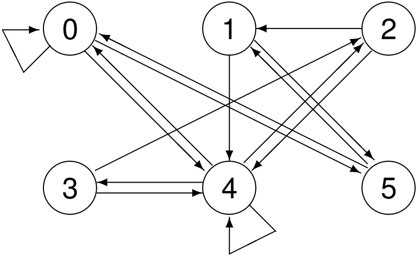


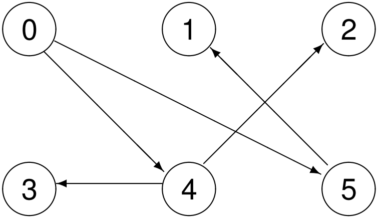


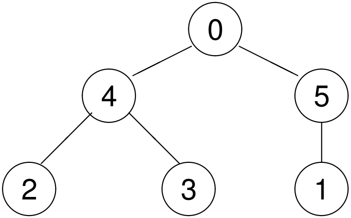


the breadth-first search tree

$$A = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$







depth-first search

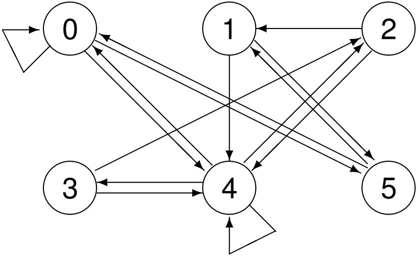
9. Consider the matrix

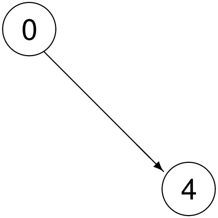
$$A = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

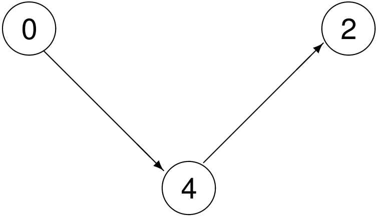
Draw the depth-first search tree for the graph, starting at vertex 0.

construction of the depth-first search tree

$$A = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

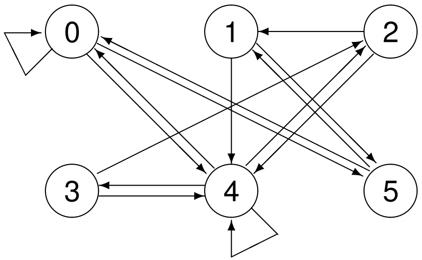


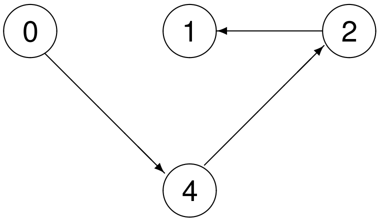


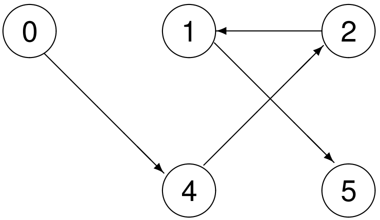


construction of the depth-first search tree

$$A = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

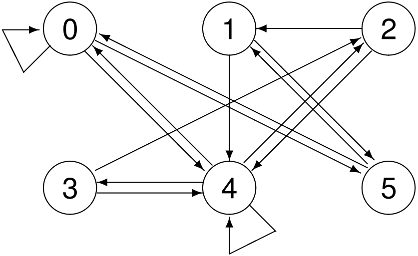


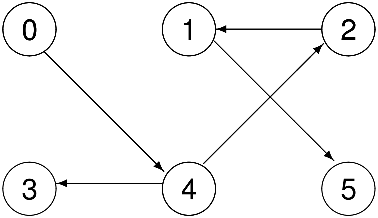


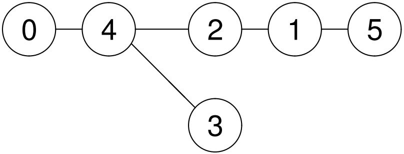


the depth-first search tree

$$A = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$







shortest paths

10. Consider the matrix

$$A = \begin{bmatrix} 0.0 & 7.3 & 6.4 & 0.0 & 0.0 & 0.3 \\ 0.0 & 0.0 & 1.4 & 4.5 & 0.0 & 7.4 \\ 0.0 & 0.0 & 1.4 & 3.9 & 2.4 & 0.0 \\ 8.3 & 0.0 & 3.9 & 2.5 & 9.0 & 0.0 \\ 0.0 & 0.0 & 8.9 & 1.4 & 1.4 & 0.0 \\ 0.0 & 2.4 & 2.0 & 0.0 & 0.0 & 1.2 \end{bmatrix}.$$

For the weighted graph defined by the adjacency matrix A, compute all shortest paths starting from the first vertex. Show all steps in the execution of the algorithm.

running Dijkstra's algorithm starting at vertex 0

$$A = \begin{bmatrix} 0.0 & 7.3 & 6.4 & 0.0 & 0.0 & 0.3 \\ 0.0 & 0.0 & 1.4 & 4.5 & 0.0 & 7.4 \\ 0.0 & 0.0 & 1.4 & 3.9 & 2.4 & 0.0 \\ 8.3 & 0.0 & 3.9 & 2.5 & 9.0 & 0.0 \\ 0.0 & 0.0 & 8.9 & 1.4 & 1.4 & 0.0 \\ 0.0 & 2.4 & 2.0 & 0.0 & 0.0 & 1.2 \end{bmatrix}$$

v	d[v]	p[v]
1	7.3	0
2	6.4	0
3	∞	0
4	∞	0
5	0.3	0

$S = \{0\}$

v	d[v]	p[v]
1	2.7	5
2	2.3	5
3	∞	0
4	∞	0
5	0.3	0

$S = \{0, 5\}$

v	d[v]	p[v]
1	2.7	5
2	2.3	5
3	6.2	2
4	4.7	2
5	0.3	0

$S = \{0, 2, 5\}$

running Dijkstra's algorithm continued

$$A = \begin{bmatrix} 0.0 & 7.3 & 6.4 & 0.0 & 0.0 & 0.3 \\ 0.0 & 0.0 & 1.4 & 4.5 & 0.0 & 7.4 \\ 0.0 & 0.0 & 1.4 & 3.9 & 2.4 & 0.0 \\ 8.3 & 0.0 & 3.9 & 2.5 & 9.0 & 0.0 \\ 0.0 & 0.0 & 8.9 & 1.4 & 1.4 & 0.0 \\ 0.0 & 2.4 & 2.0 & 0.0 & 0.0 & 1.2 \end{bmatrix}$$

v	d[v]	p[v]
1	2.7	5
2	2.3	5
3	6.2	2
4	4.7	2
5	0.3	0

$S = \{0, 1, 2, 5\}$

v	d[v]	p[v]
1	2.7	5
2	2.3	5
3	6.1	4
4	4.7	2
5	0.3	0

$S = \{0, 1, 2, 4, 5\}$

v	d[v]	p[v]
1	2.7	5
2	2.3	5
3	6.1	4
4	4.7	2
5	0.3	0

$S = \{0, 1, 2, 3, 4, 5\}$

all shortest paths from the first vertex

$$A = \begin{bmatrix} 0.0 & 7.3 & 6.4 & 0.0 & 0.0 & 0.3 \\ 0.0 & 0.0 & 1.4 & 4.5 & 0.0 & 7.4 \\ 0.0 & 0.0 & 1.4 & 3.9 & 2.4 & 0.0 \\ 8.3 & 0.0 & 3.9 & 2.5 & 9.0 & 0.0 \\ 0.0 & 0.0 & 8.9 & 1.4 & 1.4 & 0.0 \\ 0.0 & 2.4 & 2.0 & 0.0 & 0.0 & 1.2 \end{bmatrix}$$

v	d[v]	p[v]
1	2.7	5
2	2.3	5
3	6.1	4
4	4.7	2
5	0.3	0

```
The shortest paths from 0 :
1 <- 5 <- 0 : 2.7 : 2.7
2 <- 5 <- 0 : 2.3 : 2.3
3 <- 4 <- 2 <- 5 <- 0 : 6.1 : 6.1
4 <- 2 <- 5 <- 0 : 4.7 : 4.7
5 <- 0 : 0.3 : 0.3
```

minimum spanning tree

11. Consider the matrix

$$A = \begin{bmatrix} 8.7 & 1.2 & 0.0 & 0.0 & 7.6 & 2.5 \\ 1.2 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 5.7 & 3.2 \\ 0.0 & 0.0 & 0.0 & 0.0 & 5.2 & 0.0 \\ 7.6 & 0.0 & 5.7 & 5.2 & 0.0 & 9.6 \\ 2.5 & 0.0 & 3.2 & 0.0 & 9.6 & 3.0 \end{bmatrix}.$$

For the weighted graph defined by the adjacency matrix A, compute the minimum spanning tree, starting from the first vertex. Show all steps in the execution of the algorithm.

running Prim's algorithm, starting at vertex 0

$$A = \begin{bmatrix} 8.7 & 1.2 & 0.0 & 0.0 & 7.6 & 2.5 \\ 1.2 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 5.7 & 3.2 \\ 0.0 & 0.0 & 0.0 & 0.0 & 5.2 & 0.0 \\ 7.6 & 0.0 & 5.7 & 5.2 & 0.0 & 9.6 \\ 2.5 & 0.0 & 3.2 & 0.0 & 9.6 & 3.0 \end{bmatrix}$$

v	d[v]	p[v]
1	1.2	0
2	∞	0
3	∞	0
4	7.6	0
5	2.5	0

$S = \{0\}$

v	d[v]	p[v]
1	1.2	0
2	∞	0
3	∞	0
4	7.6	0
5	2.5	0

$S = \{0, 1\}$

v	d[v]	p[v]
1	1.2	0
2	3.2	5
3	∞	0
4	7.6	0
5	2.5	0

$S = \{0, 1, 5\}$

running Prim's algorithm continued

$$A = \begin{bmatrix} 8.7 & 1.2 & 0.0 & 0.0 & 7.6 & 2.5 \\ 1.2 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 5.7 & 3.2 \\ 0.0 & 0.0 & 0.0 & 0.0 & 5.2 & 0.0 \\ 7.6 & 0.0 & 5.7 & 5.2 & 0.0 & 9.6 \\ 2.5 & 0.0 & 3.2 & 0.0 & 9.6 & 3.0 \end{bmatrix}$$

v	d[v]	p[v]
1	1.2	0
2	3.2	5
3	∞	0
4	7.6	0
5	2.5	0

$S = \{0, 1, 2, 5\}$

v	d[v]	p[v]
1	1.2	0
2	3.2	5
3	5.2	4
4	7.6	0
5	2.5	0

$S = \{0, 1, 2, 4, 5\}$

v	d[v]	p[v]
1	1.2	0
2	3.2	5
3	5.2	4
4	7.6	0
5	2.5	0

$S = \{0, 1, 2, 3, 4, 5\}$

the minimum spanning tree

$$A = \begin{bmatrix} 8.7 & 1.2 & 0.0 & 0.0 & 7.6 & 2.5 \\ 1.2 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 5.7 & 3.2 \\ 0.0 & 0.0 & 0.0 & 0.0 & 5.2 & 0.0 \\ 7.6 & 0.0 & 5.7 & 5.2 & 0.0 & 9.6 \\ 2.5 & 0.0 & 3.2 & 0.0 & 9.6 & 3.0 \end{bmatrix}$$

v	d[v]	p[v]
1	1.2	0
2	3.2	5
3	5.2	4
4	7.6	0
5	2.5	0

```
The minimum spanning tree :
0
  1
  5
    2
  4
    3
```