Mathematics for Computer Science, CM0167,
Example class, Week 7,

SOLUTIONS

Dr David Marshall

1. *Consider the following table of average capacities of communication links in a computer network:*

Vertices	1	2	3	4	5	6	7
1	–	*5*	–	*2*	–	–	–
2	*4*	–	*2*	*3*	–	–	–
3	–	*3*	–	–	–	*3*	–
4	*2*	*5*	–	–	*1*	–	–
5	–	–	–	*3*	–	*4*	*5*
6	–	–	*3*	–	*4*	–	*2*
7	–	–	–	–	*4*	*2*	–

   (a) *Represent the above table as* digraph *of the computer network?*

   (b) *Using* Djikstra's algorithm*, Find the shortest path from vertex 1 to all other vertices. Express your solution as a shortest path tree.*

   (c) *Write down the* routing table *for vertex 1.*

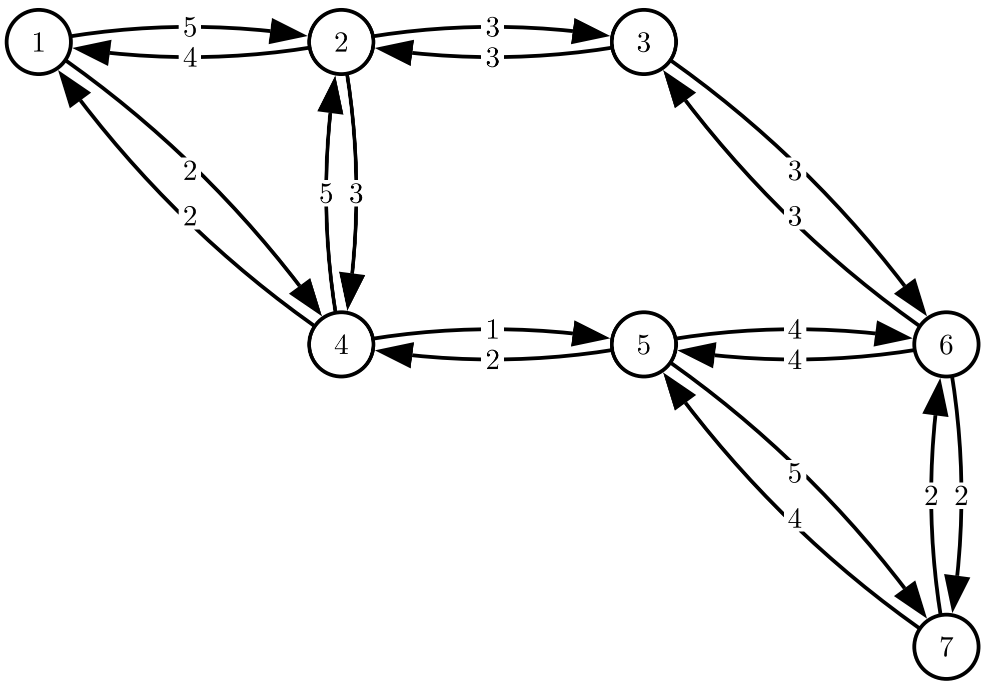
   (d) *Do the same as (a), (b) and (c) for vertex 2* etc.

   (e) *Suppose the delay weight for vertex 2 to vertex 4 decreases from 3 to 1. How does this change the shortest path tree for vertex 2?*

   (f) *If the links between vertex 5 and 6 go down what happens to the shortest path trees and routing tables for vertices 1 and 2?*

(a) Represent the above table as *digraph* of the computer network?

Graphs is:



(b) Shortest Path for vertex 1 to all other nodes

Applying Dijkstra's algorithm we get the following table:

Step	Vertex to be marked	Current potential	Distance to Vertex 1	2	3	4	5	6	7	Unchosen vertices
Step 1	1	0	0	5	–	**2**	–	–	–	2,3,4,5,6,7
Step 2	4	2	0	**5**	–	**2**	**3**	–	–	2,3,5,6,7
Step 3	5	3	0	**5**	–	**2**	**3**	7	8	2,3,6,7
Step 4	2	5	0	**5**	8	**2**	**3**	**7**	8	3,6,7
Step 5	6	7	0	**5**	**8**	**2**	**3**	**7**	**8**	3,7
Step 6	7	8	0	**5**	**8**	**2**	**3**	**7**	**8**	3

This is explained as follows (**= choose**, **= chosen**, **= dont overwrite**):

Step 1 — Only Valid Paths are to vertices 2 and 4. Path to **4** is clearly lowest.

Step 2 — Update Current Potential (2). 4 can link to 2 but this potential, 7, is **more** than current potential 5 **so dont replace**. Add a potential to Vertex **5**, this is lowest so **choose**.

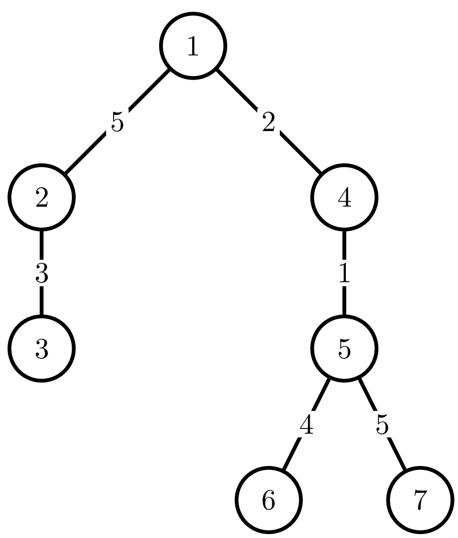
Step 3 — Update Current Potential (3). Can add paths to vertices 6 and 7 now. However Previous weight to **2** is lowest.

Step 4 — Update Current Potential (5). Can add path to vertex 3. Vertex 4 already chosen and weight is **more** for 2. Vertex **6** is now the lowest.

Step 5 — Update Current Potential (7). Could add a path from vertex 6 to vertex 3 but **weight, 10,** is more than current weight, 8. Also path from vertex 6 to vertex 7 **weight, 9,** is more than current weight, 8. Could choose either vertex 3 or 7 as weights the same, We'll choose **vertex 7**

Step 6 — Update Current Potential (7). Cant update any values paths to vertices 5 and 6 **more expensive**. Only **vertex 3 left to choose**.

We can therefore construct the following tree, by following back shortest paths from each vertex to vertex according to Dijkstra's algorithm, to show how we go from vertex one to other nodes:



(c) Routing Table for vertex 1

Therefore we can construct a *routing table* for vertex 1 as follows:

Destination	2	3	4	5	6	7
Next Node	2	2	4	4	4	4

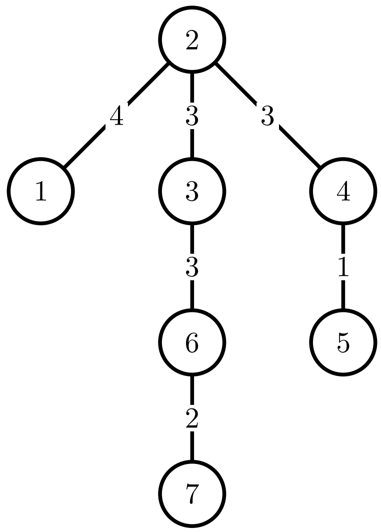
Note: All the table essentially says is which vertex to visit next to get to destination

(d) Shortest Path for vertex 2 to all other nodes

Applying Dijkstra's algorithm, similarly to Part 1, we get the following table:

Step	Vertex to be marked	Current potential	Distance to Vertex 1	2	3	4	5	6	7	Unchosen vertices
Step 1	2	0	4	0	**3**	3	–	–	–	1,3,4,5,6,7
Step 2	3	3	4	0	**3**	**3**	–	6	–	1,4,5,6,7
Step 3	4	3	4	0	**3**	**3**	**4**	6	–	1,5,6,7
Step 4	5	4	**4**	0	**3**	**3**	**4**	**6**	9	1,6,7
Step 5	1	4	**4**	0	**3**	**3**	**4**	**6**	9	6,7
Step 6	6	6	**4**	0	**3**	**3**	**4**	**6**	**8**	7

We can therefore construct a tree, as in Part 1 algorithm, to show how we go from vertex one to other nodes:



Routing Table for vertex 2

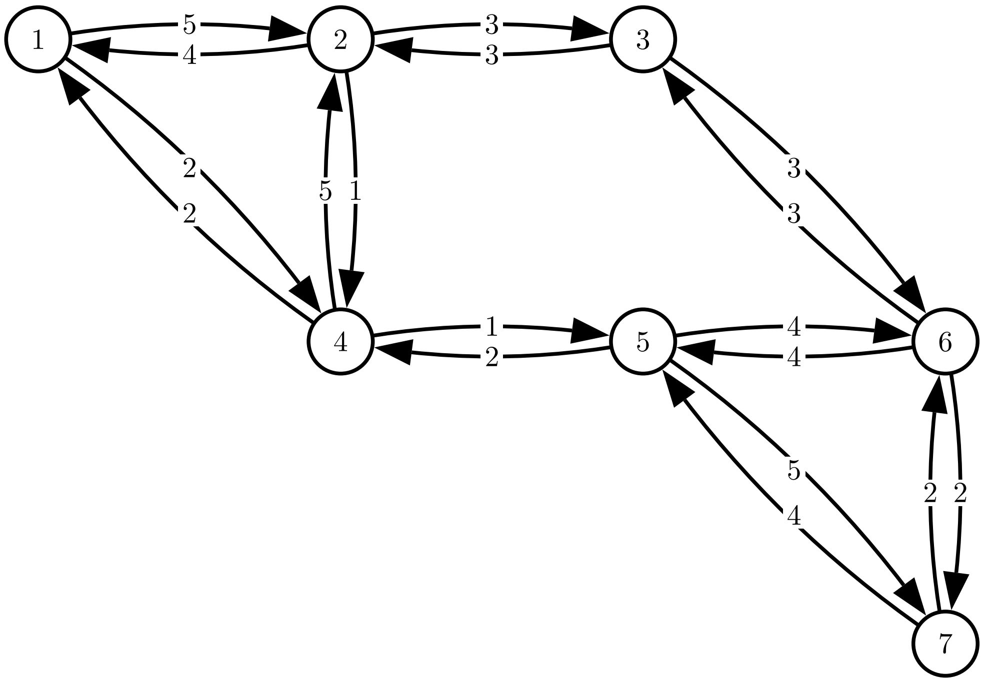
Therefore we can construct a *routing table*, as in Part 2, for vertex 2 as follows:

Destination	1	3	4	5	6	7
Next Node	1	3	4	4	3	3

(e) Weight change from vertex 2 to 4

Suppose the delay weight for vertex 2 to vertex 4 decreases from 3 to 1. How does this change the shortest path tree for vertex 2?

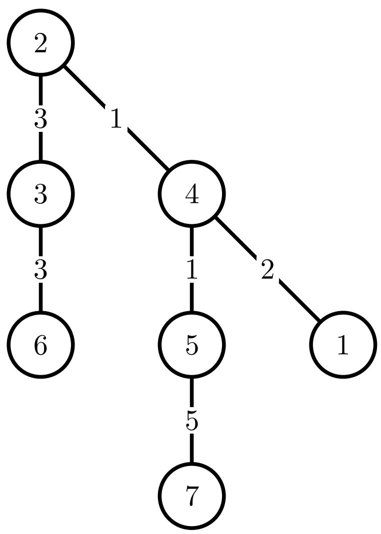
New graph:



Need to rerun Dijkstra's algorithm (Please work through the steps yourself) and we get the table:

Step	Vertex to be marked	Current potential	Distance to Vertex 1	2	3	4	5	6	7	Unchosen vertices
Step 1	2	0	4	0	3	**1**	–	–	–	1,3,4,5,6,7
Step 2	4	1	3	0	3	**1**	2	–	–	1,3,5,6,7
Step 3	5	2	3	0	**3**	**1**	**2**	6	7	1,3,6,7
Step 4	3	3	**3**	0	**3**	**3**	**4**	6	7	1,6,7
Step 5	1	3	**3**	0	**3**	**3**	**4**	**6**	7	6,7
Step 6	6	6	**4**	0	**3**	**3**	**4**	**6**	**7**	7

We can therefore construct a tree, as in Part 1 algorithm, to show how we go from vertex one to other nodes:



Therefore we can construct a *routing table*, **different from Part 3**, for vertex 2 as follows:

Destination	1	3	4	5	6	7
Next Node	4	3	4	4	3	4

(f) Links between nodes 5 and 6

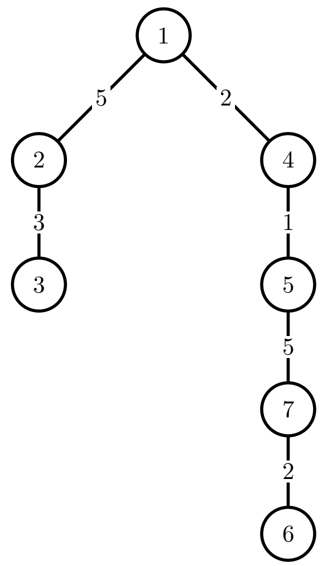
If the links between nodes 5 and 6 go down what happens to the shortest path trees and routing tables for nodes 1 and 2?

Link 56 is **not** used in for routing in vertex 2 so there is no change to the tree or the routing table.

For vertex 1, all we need to do is find the new shortest path route from node 1 to node 6. Applying Dijktra's algorithm again we can find path is 1, 4, 5, 7, 6.

Step	Vertex to be marked	Current potential	Distance to Vertex 1	2	3	4	5	6	7	Unchosen vertices
Step 1	1	0	0	5	–	**2**	–	–	–	2,3,4,5,6,7
Step 2	4	2	0	**5**	–	**2**	**3**	–	–	2,3,5,6,7
Step 3	5	3	0	**5**	–	**2**	**3**	–	8	2,3,6,7
Step 4	2	5	0	**5**	8	**2**	**3**	–	**8**	3,6,7
Step 5	7	8	0	**5**	**8**	**2**	**3**	10	**8**	3,6
Step 6	3	8	0	**5**	**8**	**2**	**3**	**10**	**8**	6

So the new tree is



The *routing table* for vertex 1 is unchanged of course:

Destination	2	3	4	5	6	7
Next Node	2	2	4	4	4	4

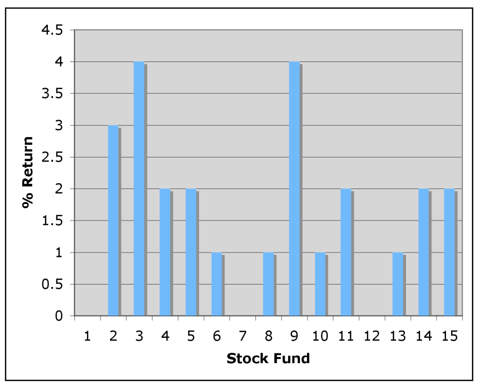
2. *We are given a sample of size 15 about the load of stock funds. The sample is*

(0%, 3%, 4%, 2%, 2%, 1%, 0%, 1%, 4%, 1%, 2%, 0%, 1%, 2%, 2%)

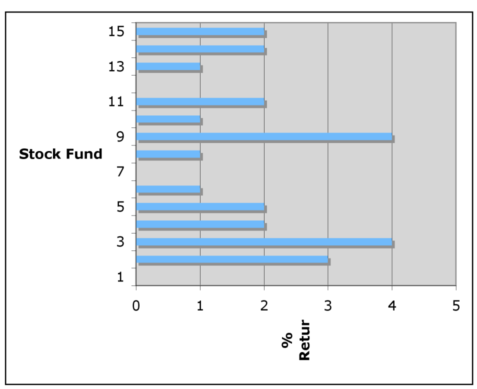
   a) *Draw a vertical bar graph, a horizontal bar graph and a circle diagram of the sample.*

   b) *Calculate the absolute and relative frequency of each load.*

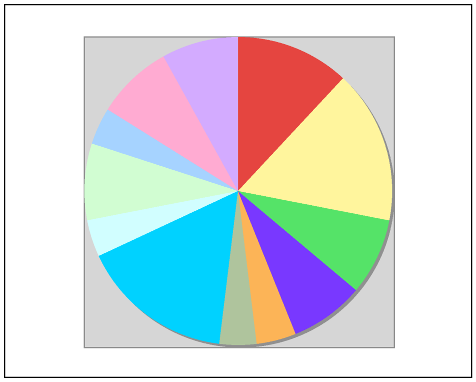
a) Vertical Bar Graph:



Horizontal Bar Graph



Circle diagram/Pie Chart:



b) absolute and relative frequency of each load

value a_j	absolute frequency n_j	relative frequency r_j
0%	3	0.2
1%	4	0.267
2%	5	0.333
3%	1	0.067
4%	2	0.133