#### Problem 1

Step 1: Array of distances

	h	u	a	m	p	f	e	s
h	0	3	4	inf	inf	inf	inf	inf
u	inf	0	6	1	6	inf	inf	inf
a	inf	inf	0	2	7	inf	inf	inf
m	inf	inf	inf	0	2	5	5	inf
p	inf	inf	inf	inf	0	3	4	inf
f	inf	inf	inf	inf	inf	0	inf	2
e	inf	inf	inf	inf	inf	inf	0	3
S	inf	0						

# Priority Queue:

V	Н	U	A	M	P	F	Е	S
D[v]	0	inf						
Prev[v]	null							
Visited	false							

Step 2:
Update adjacent edges to the initial vertex and updating H as being visited

V	Н	U	A	M	P	F	Е	S
D[v]	0	3	4	inf	inf	inf	inf	inf
Prev[v]	null	Н	Н	null	null	null	null	null
Visited	true	false						

U is minimum now

Step 3: Update adjacent edges to the initial vertex and updating U as being visited

V	Н	U	A	M	P	F	Е	S
D[v]	0	3	4	4	9	inf	inf	inf
Prev[v]	null	Н	Н	U	U	null	null	null
Visited	True	True	false	false	false	false	false	false

Step 4: Update adjacent edges to the initial vertex and updating A as being visited

V	Н	U	A	M	P	F	Е	S
D[v]	0	3	4	4	9	inf	inf	inf
Prev[v]	null	Н	Н	U	U	null	null	null
Visited	True	True	True	false	false	false	false	false

Step 5: Update adjacent edges to the initial vertex and updating M as being visited

V	Н	U	A	M	P	F	Е	S
D[v]	0	3	4	4	9	9	9	inf
Prev[v]	null	Н	Н	U	M	M	M	null
Visited	True	True	True	True	false	false	false	false

Step 6: Update adjacent edges to the initial vertex and updating P as being visited

V	Н	U	A	M	P	F	Е	S
D[v]	0	3	4	4	9	9	9	inf
Prev[v]	null	Н	Н	U	M	M	M	null
Visited	True	True	True	True	True	false	false	false

Step 7: Update adjacent edges to the initial vertex and updating F as being visited

V	Н	U	A	M	P	F	Е	S
D[v]	0	3	4	4	9	9	9	11
Prev[v]	null	Н	Н	U	M	M	M	F
Visited	True	True	True	True	True	True	false	false

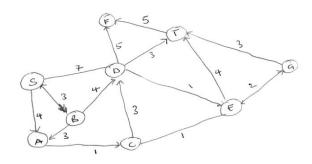
Step 8: Update adjacent edges to the initial vertex and updating E as being visited

V	Н	U	A	M	P	F	Е	S
D[v]	0	3	4	4	9	9	9	11
Prev[v]	null	Н	Н	U	M	M	M	F
Visited	True	false						

Minimum Path: F-M-U-H

### Problem 2

Problem 2.



Note: Not drawn to Scale.

# Problem 3

Visited         False         <										
Prev[v]         null         s         S         A         B <td>V</td> <td>S</td> <td>A</td> <td>В</td> <td>С</td> <td>D</td> <td>Е</td> <td>F</td> <td>G</td> <td>T</td>	V	S	A	В	С	D	Е	F	G	T
Visited         False         <	D[v]	0	4	3	-1	7	-1	-1	-1	-1
V         S         A         B         C         D         E         F         G         T           D[v]         0         4         3         -1         7         -1	Prev[v]	null								
D[v]         0         4         3         -1         7         -1         -1         -1         -1         -1           Prev[v]         null         S         S         null         S         null	Visited	False								
D[v]         0         4         3         -1         7         -1         -1         -1         -1         -1           Prev[v]         null         S         S         null         S         null										
Prev[v]         null         S         S         null         S         null         null<	V	S	A	В	С	D	Е	F	G	T
Visited True False	D[v]	0	4	3	-1	7	-1	-1	-1	-1
V         S         A         B         C         D         E         F         G         T           D[v]         0         4         3         -1         7         -1	Prev[v]	null	S	S	null	S	null	null	null	null
D[v]         0         4         3         -1         7         -1         -1         -1         -1           Prev[v]         null         S         S         null         B         null         null <td>Visited</td> <td>True</td> <td>False</td> <td>False</td> <td>False</td> <td>False</td> <td>False</td> <td>False</td> <td>False</td> <td>False</td>	Visited	True	False							
D[v]         0         4         3         -1         7         -1         -1         -1         -1           Prev[v]         null         S         S         null         B         null         null <td></td> <td></td> <td>•</td> <td></td> <td></td> <td>•</td> <td></td> <td></td> <td></td> <td></td>			•			•				
Prev[v]       null       S       S       null       B       null	V	S	A	В	С	D	Е	F	G	T
Visited True False True False	D[v]	0	4	3	-1	7	-1	-1	-1	-1
V         S         A         B         C         D         E         F         G         T           D[v]         0         4         3         5         7         -1         -1         -1         -1           Prev[v]         null         S         S         A         B         null         null         null         null	Prev[v]	null	S	S	null	В	null	null	null	null
D[v]         0         4         3         5         7         -1         -1         -1         -1           Prev[v]         null         S         S         A         B         null         null         null         null	Visited	True	False	True	False	False	False	False	False	False
D[v]         0         4         3         5         7         -1         -1         -1         -1           Prev[v]         null         S         S         A         B         null         null         null         null										
Prev[v] null S S A B null null null nul	V	S	A	В	С	D	Е	F	G	T
	D[v]	0	4	3	5	7	-1	-1	-1	-1
Visited True True False False False False False False False	Prev[v]	null	S	S	A	В	null	null	null	null
	Visited	True	True	True	False	False	False	False	False	False
			ı	ı	ı	ı	ı	ı	T	1
V S A B C D E F G T	V	S	A	В	С	D	Е	F	G	Т
D[v] 0 4 3 5 7 10 -1 -1	D[v]	0	4	3	5	7	10	-1	-1	-1
Prev[v] null S S A B C null null nul	Prev[v]	null	S	S	A	В	С	null	null	null
Visited     True     True     True     False     False     False     False     False	Visited	True	True	True	True	False	False	False	False	False

V	S	A	В	С	D	Е	F	G	Т
D[v]	0	4	3	5	7	10	15	-1	13
Prev[v]	null	S	S	A	В	С	D	null	D
Visited	True	True	True	True	True	False	False	False	False
V	S	A	В	C	D	Е	F	G	T
D[v]	0	4	3	5	7	10	15	21	13
Prev[v]	null	S	S	A	В	С	D	Е	D
Visited	True	True	True	True	True	True	False	False	False
V	S	A	В	С	D	Е	F	G	Т
D[v]	0	4	3	5	7	10	15	21	13
Prev[v]	null	S	S	A	В	С	D	Е	D
Visited	True	True	True	True	True	True	False	False	True
	•	•	•	•	•	•	•	•	•
V	S	A	В	С	D	Е	F	G	Т
D[v]	0	4	3	5	7	10	15	21	13
Prev[v]	null	S	S	A	В	С	D	Е	D
Visited	True	True	True	True	True	True	True	False	True
V	S	A	В	С	D	Е	F	G	Т
D[v]	0	4	3	5	7	10	15	21	13
Prev[v]	null	S	S	A	В	С	D	Е	D
Visited	True	True	True	True	True	True	True	True	True

Final minimum path using Dijkstra's algorithm is S-B-D-T

Problem 4

Matrix when path is undirected

	S	A	В	С	D	Е	F	G	Т
S	0	4	3	-1	7	-1	-1	-1	-1
A	4	0	-1	1	-1	-1	-1	-1	-1
В	3	-1	0	-1	4	-1	-1	-1	-1
С	-1	1	-1	0	3	1	-1	-1	-1
D	7	-1	4	3	0	1	5	-1	3
Е	-1	-1	-1	-1	-1	0	-1	2	4
F	-1	-1	-1	-1	5	-1	0	-1	-1
G	-1	-1	-1	-1	-1	2	-1	0	3
Т	-1	-1	-1	-1	3	4	-1	3	0

#### Problem 5

It is impossible to work with negative cost in Dijkstra's algorithm because with negative costs, it is likely the path with the minimum cost will be infinity.

#### Problem 6

To find the largest-cost path, we would have to negate all the cost values on each path. Doing so will make it impossible to use the Dijkstra's algorithm since it is not applicable to negative costs.