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 | inf | inf | inf | inf | inf | inf | 0 |

Priority Queue:

| V | H | U | A | M | P | F | E | S |
|---|---|---|---|---|---|---|---|---|
| D[v] | 0 | inf | inf | inf | inf | inf | inf | inf |
| Prev[v] | null | null | null | null | null | null | null | null |
| Visited | false | false | false | false | false | false | false | false |

Step 2:

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

| V | H | U | A | M | P | F | E | S |
|---|---|---|---|---|---|---|---|---|
| D[v] | 0 | 3 | 4 | inf | inf | inf | inf | inf |
| Prev[v] | null | H | H | null | null | null | null | null |
| Visited | true | false | false | false | false | false | false | false |

U is minimum now

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

| V | H | U | A | M | P | F | E | S |
|---|---|---|---|---|---|---|---|---|
| D[v] | 0 | 3 | 4 | 4 | 9 | inf | inf | inf |
| Prev[v] | null | H | H | 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 | H | U | A | M | P | F | E | S |
|---|---|---|---|---|---|---|---|---|
| D[v] | 0 | 3 | 4 | 4 | 9 | inf | inf | inf |
| Prev[v] | null | H | H | 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 | H | U | A | M | P | F | E | S |
|---|---|---|---|---|---|---|---|---|
| D[v] | 0 | 3 | 4 | 4 | 9 | 9 | 9 | inf |
| Prev[v] | null | H | H | 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 | H | U | A | M | P | F | E | S |
|---|---|---|---|---|---|---|---|---|
| D[v] | 0 | 3 | 4 | 4 | 9 | 9 | 9 | inf |
| Prev[v] | null | H | H | 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

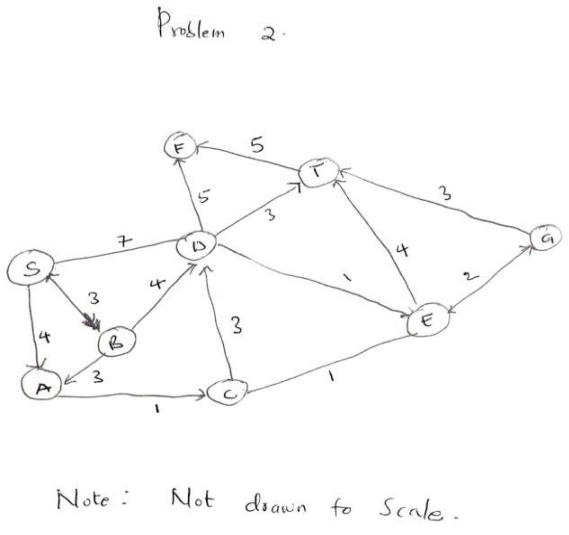
| V | H | U | A | M | P | F | E | S |
|---|---|---|---|---|---|---|---|---|
| D[v] | 0 | 3 | 4 | 4 | 9 | 9 | 9 | 11 |
| Prev[v] | null | H | H | U | M | M | M | F |
| Visited | True | True | True | True | True | True | false | false |

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

| V | H | U | A | M | P | F | E | S |
|---|---|---|---|---|---|---|---|---|
| D[v] | 0 | 3 | 4 | 4 | 9 | 9 | 9 | 11 |
| Prev[v] | null | H | H | U | M | M | M | F |
| Visited | True | True | True | True | True | True | True | false |

Minimum Path: F-M-U-H

Problem 2



Problem 3

| V | S | A | B | C | D | E | F | G | T |
|---|---|---|---|---|---|---|---|---|---|
| D[v] | 0 | 4 | 3 | -1 | 7 | -1 | -1 | -1 | -1 |
| Prev[v] | null | null | null | null | null | null | null | null | null |
| Visited | False | False | False | False | False | False | False | False | False |

| V | S | A | B | C | D | E | F | G | T |
|---|---|---|---|---|---|---|---|---|---|
| D[v] | 0 | 4 | 3 | -1 | 7 | -1 | -1 | -1 | -1 |
| Prev[v] | null | S | S | null | S | null | null | null | null |
| Visited | True | False | False | False | False | False | False | False | False |

| V | S | A | B | C | D | E | F | G | T |
|---|---|---|---|---|---|---|---|---|---|
| D[v] | 0 | 4 | 3 | -1 | 7 | -1 | -1 | -1 | -1 |
| Prev[v] | null | S | S | null | B | null | null | null | null |
| Visited | True | False | True | False | False | False | False | False | False |

| 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 |
| Visited | True | True | True | False | False | False | False | False | False |

| V | S | A | B | C | D | E | F | G | T |
|---|---|---|---|---|---|---|---|---|---|
| D[v] | 0 | 4 | 3 | 5 | 7 | 10 | -1 | -1 | -1 |
| Prev[v] | null | S | S | A | B | C | null | null | null |
| Visited | True | True | True | True | False | False | False | False | False |

| V | S | A | B | C | D | E | F | G | T |
|---|---|---|---|---|---|---|---|---|---|
| D[v] | 0 | 4 | 3 | 5 | 7 | 10 | 15 | -1 | 13 |
| Prev[v] | null | S | S | A | B | C | D | null | D |
| Visited | True | True | True | True | True | False | False | False | False |

| V | S | A | B | C | D | E | F | G | T |
|---|---|---|---|---|---|---|---|---|---|
| D[v] | 0 | 4 | 3 | 5 | 7 | 10 | 15 | 21 | 13 |
| Prev[v] | null | S | S | A | B | C | D | E | D |
| Visited | True | True | True | True | True | True | False | False | False |

| V | S | A | B | C | D | E | F | G | T |
|---|---|---|---|---|---|---|---|---|---|
| D[v] | 0 | 4 | 3 | 5 | 7 | 10 | 15 | 21 | 13 |
| Prev[v] | null | S | S | A | B | C | D | E | D |
| Visited | True | True | True | True | True | True | False | False | True |

| V | S | A | B | C | D | E | F | G | T |
|---|---|---|---|---|---|---|---|---|---|
| D[v] | 0 | 4 | 3 | 5 | 7 | 10 | 15 | 21 | 13 |
| Prev[v] | null | S | S | A | B | C | D | E | D |
| Visited | True | True | True | True | True | True | True | False | True |

| V | S | A | B | C | D | E | F | G | T |
|---|---|---|---|---|---|---|---|---|---|
| D[v] | 0 | 4 | 3 | 5 | 7 | 10 | 15 | 21 | 13 |
| Prev[v] | null | S | S | A | B | C | D | E | 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 | B | C | D | E | F | G | T |
|---|---|---|---|---|---|---|---|---|---|
| S | 0 | 4 | 3 | -1 | 7 | -1 | -1 | -1 | -1 |
| A | 4 | 0 | -1 | 1 | -1 | -1 | -1 | -1 | -1 |
| B | 3 | -1 | 0 | -1 | 4 | -1 | -1 | -1 | -1 |
| C | -1 | 1 | -1 | 0 | 3 | 1 | -1 | -1 | -1 |
| D | 7 | -1 | 4 | 3 | 0 | 1 | 5 | -1 | 3 |
| E | -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 |
| T | -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.