16 b)

Current

A = 0:

B: 5+0 = 5
F: 3+0 = 3 – Update prev node and values
D: 6+0 = 6

A is now visited.

Next node w/ shortest tentative distance

F = 3:

G = 8+3 = 11 – Update prev node and value
A is already visited.

F is now visited.

B = 5:

C: 4+5 = 9 – Update prev node and values
E: 2+5 = 7
A is already visited. B is now visited.

D = 6:

C: 3+6 = 9 – no update w/ values.
E: 1+6 = 7 Update prev node because values for distance are equivalent!
A is already visited.

D is now visited.

E = 7:

C: 1+7 = 8 – Update prev node and value
B, D, are already visited.

E is now visited.

C = 8:

G: 2+8 = 10 – Update prev node and value
B, D, E already visited.

C is now visited.

G = 10 All other nodes are visited. Algorithm ends.

Unvisited	Prev Node	Visited
~~A: 0~~	~~0/A~~	A
~~B: ∞ 5~~	A	B
~~C: ∞ 9 8~~	~~B, D,~~ E	C
~~D: ∞ 6~~	A	D
~~E: ∞ 7~~	B, D	E
~~F: ∞ 3~~	A	F
~~G: ∞ 11 10~~	~~F~~ C	G

Shortest Path:

ABECG

(found by working backwards ⇒ G, C, E (B, D), A from G)

ADECG

Shortest Distance:

G = 10

6c - E node fails

Current Node

A=0:

B: 5+0=5
F: 3+0=3 — Update
D: 6+0=6

F=3:

G: 8+3=11 - Update

B=5:

C: 4+5=9 - Update

E is not functional.

D=6:

C: 3+6=9 - Equivalent value, update prev node (add D to list)

C=9:

G: 2+9=11 - Equivalent value, update prev node (add C)

G=11: Done

Unvisited	Prev Node	Visited
A: 0	N/A	A
~~B: ∞~~ 5	A	B
~~C: ∞~~ 9	B, D	C
~~D: ∞~~ 6	A	D
~~E~~ - Fails	/	/
~~F: ∞~~ 3	A	F
~~G: ∞~~ 11	F, C	G

Working Backwards: Shortest Path

G → F → A: AFG

G → C → B → A: ABCG

G → C → D → A: ADCG

Shortest Length:

G=11

11