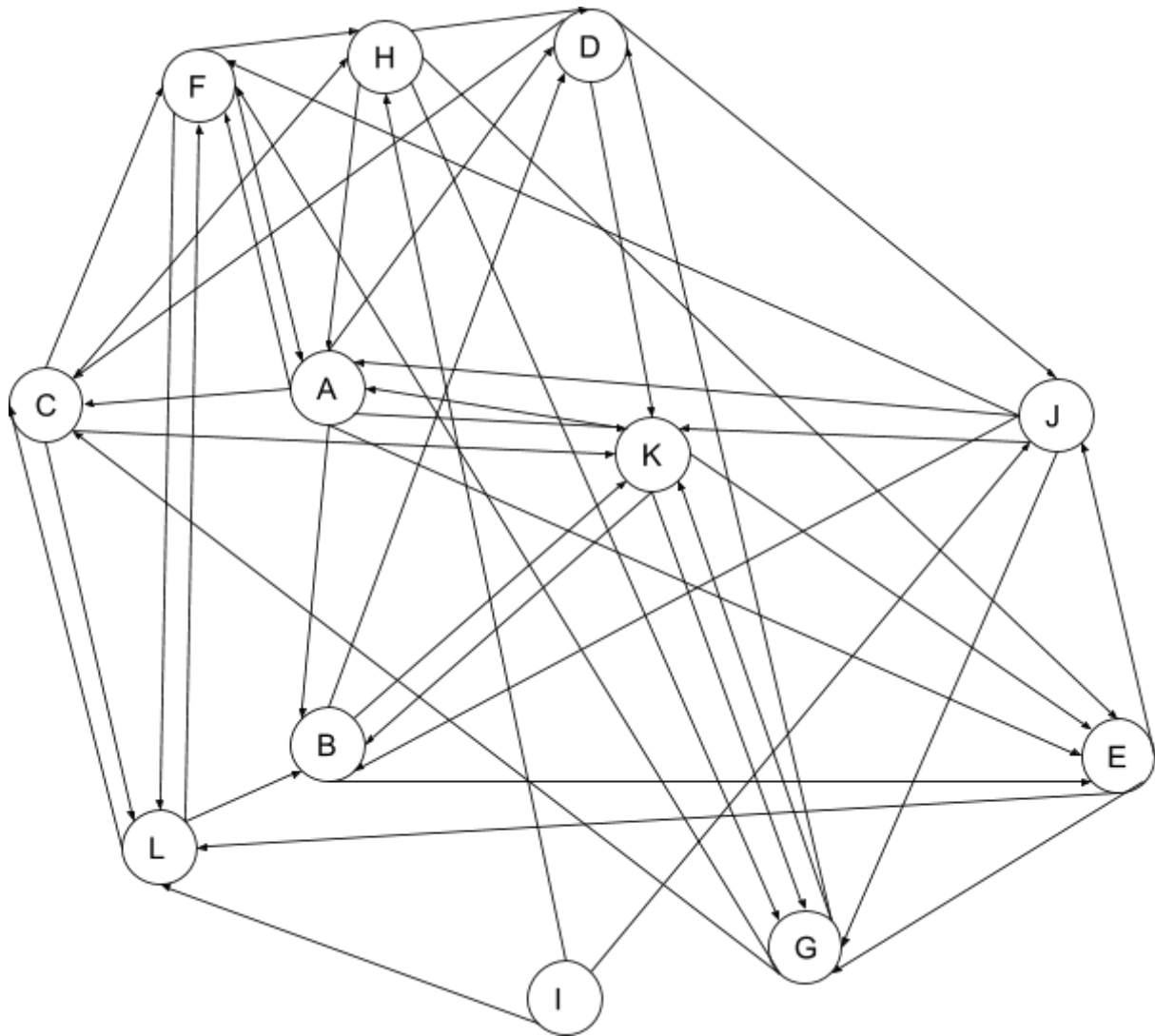


Assignment 5 Question 2

Graph 1:



1. Depth-first traversal

Event	Stack	Current Result
Visit B	B	B
Visit D	BD	BD
Visit K	BDK	BDK
Visit G	BDKG	BDKG
Visit C	BDKGC	BDKGC

Visit F	BDKGCF	BDKGCF
Visit H	BDKGCFH	BDKGCFH
Visit E	BDKGCFHE	BDKGCFHE
Visit L	BDKGCFHEL	BDKGCFHEL
Pop L	BDKGCFHE	BDKGCFHEL
Visit J	BDKGCFHEJ	BDKGCFHELJ
Visit A	BDKGCFHEJA	BDKGCFHELJA
Pop A	BDKGCFHEJ	BDKGCFHELJA
Pop J	BDKGCFHE	BDKGCFHELJA
Pop E	BDKGCFH	BDKGCFHELJA
Pop H	BDKGCF	BDKGCFHELJA
Pop F	BDKGC	BDKGCFHELJA
Pop C	BDKG	BDKGCFHELJA
Pop G	BDK	BDKGCFHELJA
Pop K	BD	BDKGCFHELJA
Pop D	B	BDKGCFHELJA
Pop B	Empty	BDKGCFHELJA

List of nodes in depth-first traversal order: BDKGCFHELJA

2. Breadth-first traversal

Event	Queue	Current Vertex	Current Result
Visit L	Empty	L	L
Visit C	C	L	LC
Visit B	CB	L	LCB
Visit F	CBF	L	LCBF
Dequeue C	BF	C	LCBF
Visit H	BFH	C	LCBFH

Visit K	BFHK	C	LCBFHK
Dequeue B	FHK	B	LCBFHK
Visit D	FHKD	B	LCBFHKD
Visit E	FHKDE	B	LCBFHKDE
Dequeue F	HKDE	F	LCBFHKDE
Visit A	HKDEA	F	LCBFHKDEA
Dequeue H	KDEA	H	LCBFHKDEA
Visit G	KDEAG	H	LCBFHKDEAG
Dequeue K	DEAG	K	LCBFHKDEAG
Dequeue D	EAG	D	LCBFHKDEAG
Visit J	EAGJ	D	LCBFHKDEAGJ
Dequeue E	AGJ	E	LCBFHKDEAGJ
Dequeue A	GJ	A	LCBFHKDEAGJ
Dequeue G	J	G	LCBFHKDEAGJ
Dequeue J	Empty	J	LCBFHKDEAGJ

List of nodes in breadth-first traversal order: LCBFHKDEAGJ

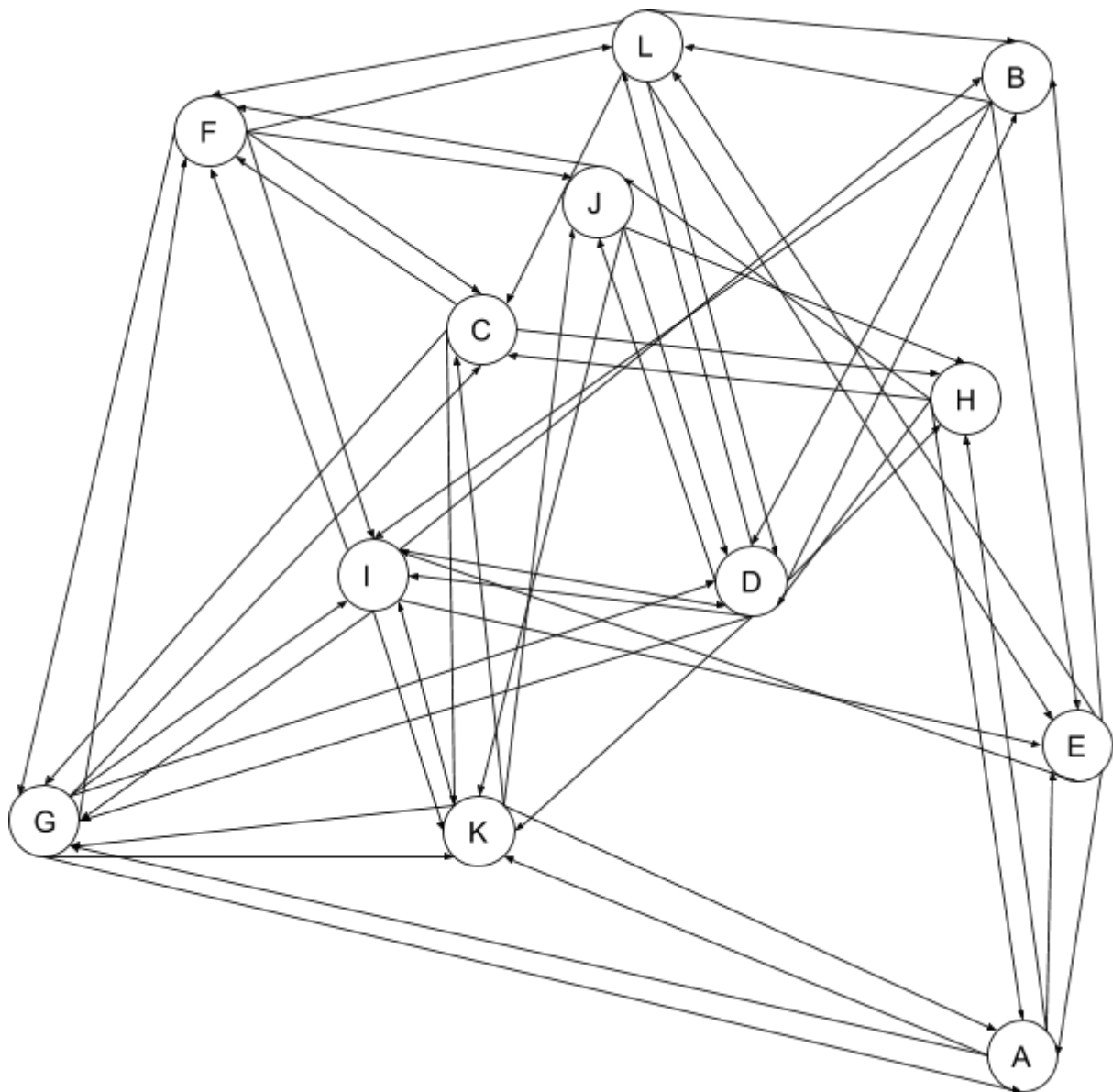
3. Dijkstra's algorithm

From H to	A	B	C	D	E	F	G	I	J	K	L
Initially	inf	inf	inf	inf	inf	inf	inf	inf	inf	inf	inf
Step 1 (From H)	60(H)	inf	inf	50(H)	10(H)	inf	70(H)	inf	inf	inf	inf
Step 2 (From E)	60(H)	inf	inf	50(H)	*10(H)	inf	50(E)	inf	30(E)	inf	30(E)
Step 3 (From J)	40(J)	90(J)	inf	50(H)	*10(H)	110(J)	40(J)	inf	*30(E)	120(J)	30(E)
Step 4 (From L)	40(J)	90(J)	90(L)	50(H)	*10(H)	110(J)	40(J)	inf	*30(E)	120(J)	*30(E)

Step 5 (From A)	*40(J)	80(A)	60(A)	50(H)	*10(H)	110(J)	40(J)	inf	*30(E)	120(J)	*30(E)
Step 6 (From G)	*40(J)	80(A)	60(A)	50(H)	*10(H)	100(G)	*40(J)	inf	*30(E)	120(J)	*30(E)
Step 7 (From D)	*40(J)	80(A)	60(A)	*50(H)	*10(H)	100(G)	*40(J)	inf	*30(E)	100(D)	*30(E)
Step 8 (From C)	*40(J)	80(A)	*60(A)	*50(H)	*10(H)	90(C)	*40(J)	inf	*30(E)	90(C)	*30(E)
Step 9 (From B)	*40(J)	*80(A)	*60(A)	*50(H)	*10(H)	90(C)	*40(J)	inf	*30(E)	85(B)	*30(E)
Step 10 (From K)	*40(J)	*80(A)	*60(A)	*50(H)	*10(H)	90(C)	*40(J)	inf	*30(E)	*85(B)	*30(E)
Step 11 (From F)	*40(J)	*80(A)	*60(A)	*50(H)	*10(H)	*90(C)	*40(J)	inf	*30(E)	*85(B)	*30(E)

Shortest path from H to:
 E is H-E, which cost 10
 J is H-E-J, which cost 30
 L is H-E-L, which cost 30
 A is H-E-J-A, which cost 40
 G is H-E-J-G, which cost 40
 D is H-D, which cost 50
 C is H-E-J-A-C, which cost 60
 B is H-E-J-A-B, which cost 80
 K is H-E-J-A-B-K, which cost 85
 F is H-E-J-A-C-F, which cost 90
 There's no path from H to I

Graph 2:



1. Depth-first traversal

Event	Stack	Current Result
Visit B	B	B
Visit L	BL	BL
Visit F	BLF	BLF
Visit G	BLFG	BLFG
Visit A	BLFGA	BLFGA
Visit K	BLFGAK	BLFGAK

Visit J	BLFGAKJ	BLFGAKJ
Visit H	BLFGAKJH	BLFGAKJH
Visit D	BLFGAKJHD	BLFGAKJHD
Visit I	BLFGAKJHDI	BLFGAKJHDI
Visit E	BLFGAKJHDIE	BLFGAKJHDIE
Pop E	BLFGAKJHDI	BLFGAKJHDIE
Pop I	BLFGAKJHD	BLFGAKJHDIE
Pop D	BLFGAKJH	BLFGAKJHDIE
Pop H	BLFGAKJ	BLFGAKJHDIE
Visit C	BLFGAKJC	BLFGAKJHDIEC
Pop C	BLFGAKJ	BLFGAKJHDIEC
Pop J	BLFGAK	BLFGAKJHDIEC
Pop K	BLFGA	BLFGAKJHDIEC
Pop A	BLFG	BLFGAKJHDIEC
Pop G	BLF	BLFGAKJHDIEC
Pop F	BL	BLFGAKJHDIEC
Pop L	B	BLFGAKJHDIEC
Pop B	Empty	BLFGAKJHDIEC

List of nodes in depth-first traversal order: BLFGAKJHDIEC

2. Breadth-first traversal

Event	Queue	Current Vertex	Current Result
Visit L	Empty	L	L
Visit C	C	L	LC
Visit B	CB	L	LCB
Visit F	CBF	L	LCBF
Visit E	CBFE	L	LCBFE

Visit D	CBFED	L	LCBFED
Dequeue C	BFED	C	LCBFED
Visit G	BFEDG	C	LCBFEDG
Visit H	BFEDGH	C	LCBFEDGH
Visit K	BFEDGHK	C	LCBFEDGHK
Dequeue B	FEDGHK	B	LCBFEDGHK
Visit I	FEDGHKI	B	LCBFEDGHKI
Dequeue F	EDGHKI	F	LCBFEDGHKI
Visit J	EDGHKIJ	F	LCBFEDGHKIJ
Dequeue E	DGHKIJ	E	LCBFEDGHKIJ
Visit A	DGHKIJA	E	LCBFEDGHKIJA
Dequeue D	GHKIJA	D	LCBFEDGHKIJA
Dequeue G	HKIJA	G	LCBFEDGHKIJA
Dequeue H	KIJA	H	LCBFEDGHKIJA
Dequeue K	IJA	K	LCBFEDGHKIJA
Dequeue I	JA	I	LCBFEDGHKIJA
Dequeue J	A	J	LCBFEDGHKIJA
Dequeue A	Empty	A	LCBFEDGHKIJA

List of nodes in breadth-first traversal order: LCBFEDGHKIJA

3. Dijkstra's algorithm

From H to	A	B	C	D	E	F	G	I	J	K	L
Initially	inf	inf	inf	inf	inf	inf	inf	inf	inf	inf	inf
Step 1 (From H)	60(H)	inf	70(H)	90(H)	inf	inf	inf	inf	20(H)	inf	inf
Step 2 (From J)	60(H)	inf	70(H)	80(J)	inf	70(J)	inf	inf	*20(H)	100(J)	inf

Step 3 (From A)	*60(H)	inf	70(H)	80(J)	130(A)	70(J)	100(A)	inf	*20(H)	90(A)	inf
Step 4 (From C)	*60(H)	inf	*70(H)	80(J)	130(A)	70(J)	90(C)	inf	*20(H)	80(C)	120(C)
Step 5 (From F)	*60(H)	inf	*70(H)	80(J)	130(A)	*70(J)	90(C)	150(F)	*20(H)	80(C)	100(F)
Step 6 (From D)	*60(H)	150(D)	*70(H)	*80(J)	130(A)	*70(J)	90(C)	110(D)	*20(H)	80(C)	90(D)
Step 7 (From K)	*60(H)	150(D)	*70(H)	*80(J)	130(A)	*70(J)	90(C)	110(D)	*20(H)	*80(C)	90(D)
Step 8 (From G)	*60(H)	150(D)	*70(H)	*80(J)	130(A)	*70(J)	*90(C)	100(G)	*20(H)	*80(C)	90(D)
Step 9 (From L)	*60(H)	130(L)	*70(H)	*80(J)	120(L)	*70(J)	*90(C)	100(G)	*20(H)	*80(C)	*90(D)
Step 10 (From I)	*60(H)	120(I)	*70(H)	*80(J)	110(I)	*70(J)	*90(C)	*100(G)	*20(H)	*80(C)	*90(D)
Step 11 (From E)	*60(H)	120(I)	*70(H)	*80(J)	*110(I)	*70(J)	*90(C)	*100(G)	*20(H)	*80(C)	*90(D)
Step 12 (From B)	*60(H)	*120(I)	*70(H)	*80(J)	*110(I)	*70(J)	*90(C)	*100(G)	*20(H)	*80(C)	*90(D)

Shortest path from H to:
 J is H-J, which cost 20
 A is H-A, which cost 60
 C is H-C, which cost 70
 F is H-J-F, which cost 70
 D is H-J-D, which cost 80
 K is H-C-K, which cost 80
 G is H-C-G, which cost 90
 L is H-J-D-L, which cost 90
 I is H-C-G-I, which cost 100
 E is H-C-G-I-E, which cost 110
 B is H-C-G-I-B, which cost 120