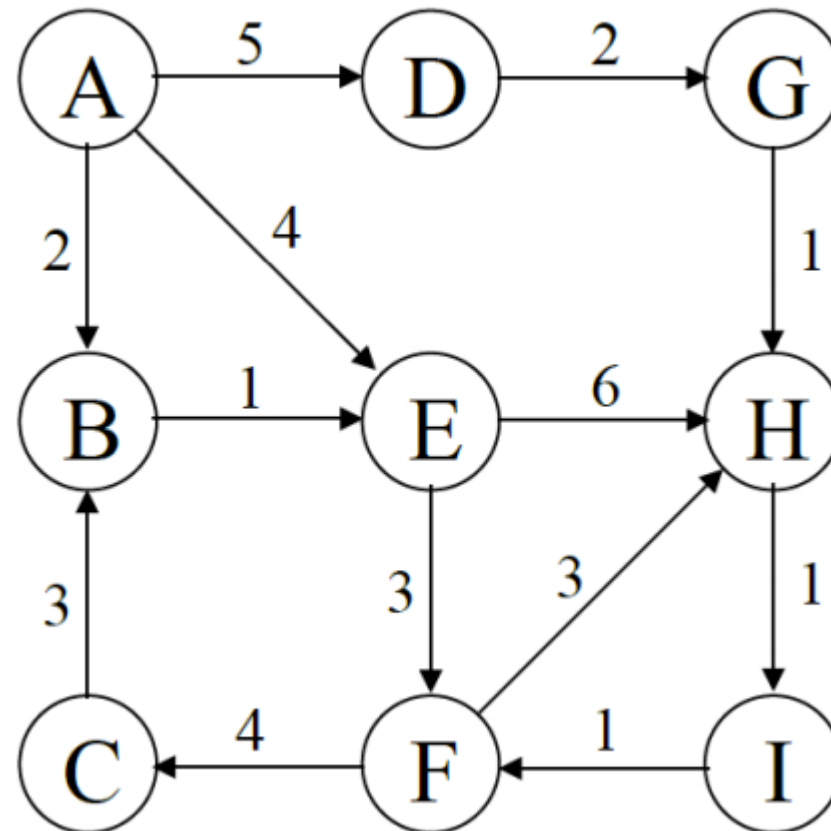


Dijkstra Algorithm

- For each vertex v , we must keep track of:
 - Whether the vertex still needs to be processed: $\text{toBeChecked}(v)$
 - The current distance from the source for the shortest path found so far: $\text{currDist}(v)$
 - The predecessor vertex for the shortest path found so far: $\text{pred}(v)$

⌈

Example: Dijkstra Algorithm



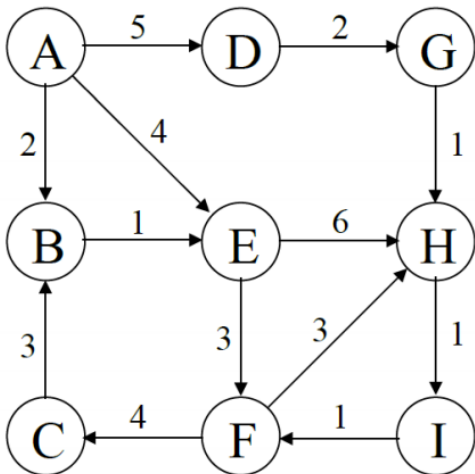


Initial set up: source vertex is A:

	toBeChecked	currDist	pred
A	t	0	\
B	t	∞	\
C	t	∞	\
D	t	∞	\
E	t	∞	\
F	t	∞	\
G	t	∞	\
H	t	∞	\
I	t	∞	\

v is vertex A

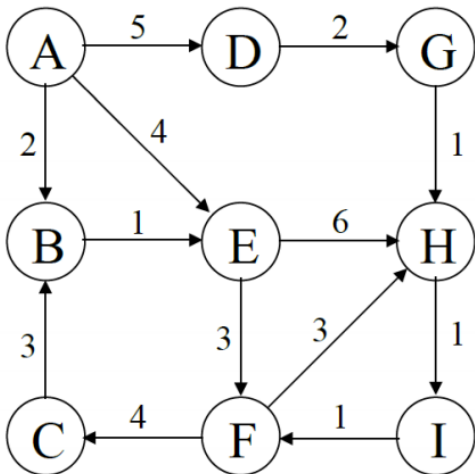
u is B, D, E



	toBeChecked	currDist	pred
A	t f	0	\
B	t	∞ 2	\ A
C	t	∞	\
D	t	∞ 5	\ A
E	t	∞ 4	\ A
F	t	∞	\
G	t	∞	\
H	t	∞	\
I	t	∞	\

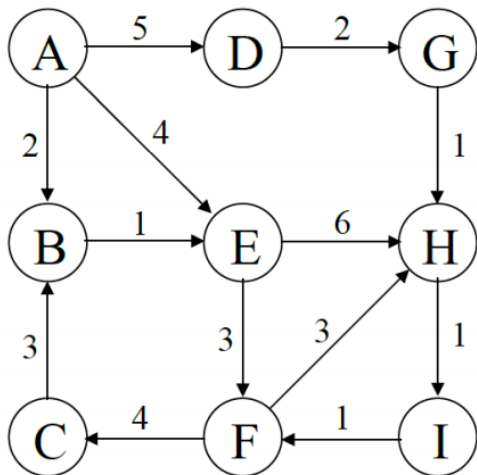
• v is vertex B

▫ u is E



	toBeChecked	currDist	pred
A	t f	0	\
B	t f	∞ 2	/ A
C	t	∞	\
D	t	∞ 5	/ A
E	t	∞ 4 3	/ A B
F	t	∞	\
G	t	∞	\
H	t	∞	\
I	t	∞	\

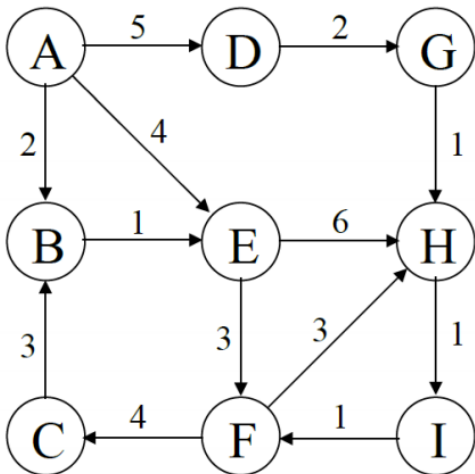
- v is vertex E
 - u is F, H



	toBeChecked	currDist	pred
A	t f	0	\
B	t f	∞ 2	/ A
C	t	∞	\
D	t	∞ 5	/ A
E	t f	∞ 4 3	/ A B
F	t	∞ 6	/ E
G	t	∞	\
H	t	∞ 9	/ E
I	t	∞	\

v is vertex D

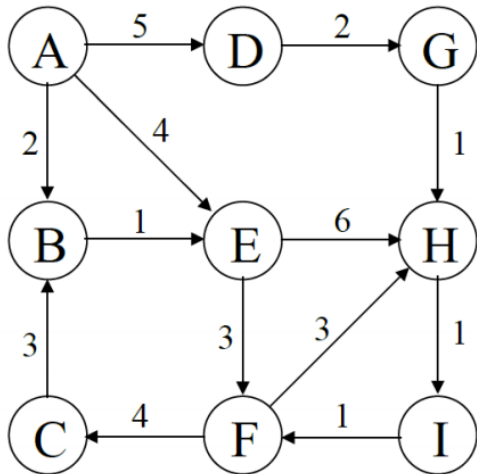
u is G



	toBeChecked	currDist	pred
A	t f	0	\
B	t f	∞ 2	/ A
C	t	∞	\
D	t f	∞ 5	/ A
E	t f	∞ 4 3	/ A B
F	t	∞ 6	/ E
G	t	∞ 7	/ D
H	t	∞ 9	/ E
I	t	∞	\

v is vertex F

u is C, H

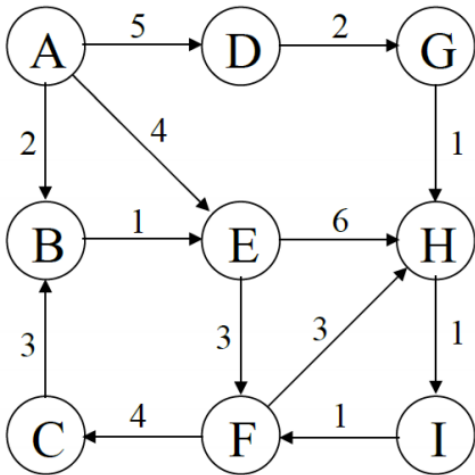


	toBeChecked	currDist	pred
A	t f	0	\
B	t f	∞ 2	/ A
C	t	∞ 10	/ F
D	t f	∞ 5	/ A
E	t f	∞ 4 3	/ A B
F	t f	∞ 6	/ E
G	t	∞ 7	/ D
H	t	∞ 9	/ E
I	t	∞	\



v is vertex G

□ u is H

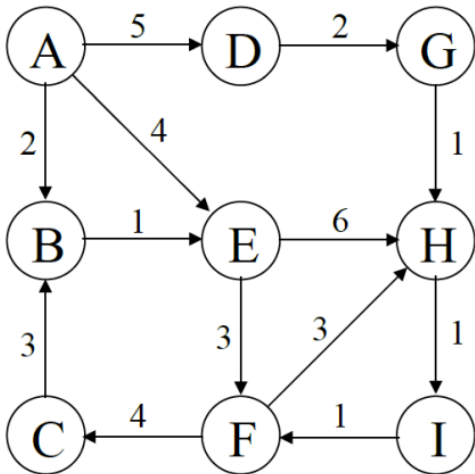


	toBeChecked	currDist	pred
A	t f	0	\
B	t f	∞ 2	/ A
C	t	∞ 10	/ F
D	t f	∞ 5	/ A
E	t f	∞ 4 3	/ A B
F	t f	∞ 6	/ E
G	t f	∞ 7	/ D
H	t	∞ 9 8	/ E G
I	t	∞	\



v is vertex H

□ u is I

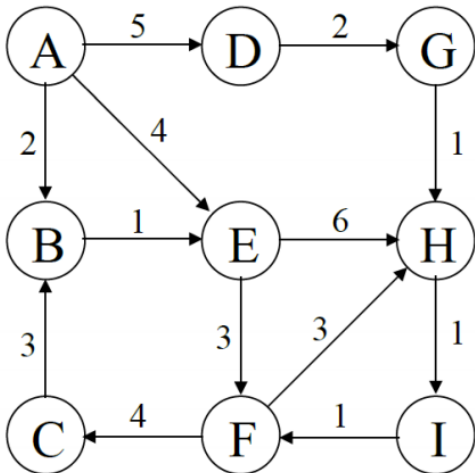


	toBeChecked	currDist	pred
A	t f	0	\
B	t f	∞ 2	/ A
C	t	∞ 10	/ F
D	t f	∞ 5	/ A
E	t f	∞ 4 3	/ A B
F	t f	∞ 6	/ E
G	t f	∞ 7	/ D
H	t f	∞ 9 8	/ E G
I	t	∞ 9	/ H



v is vertex I

u is empty

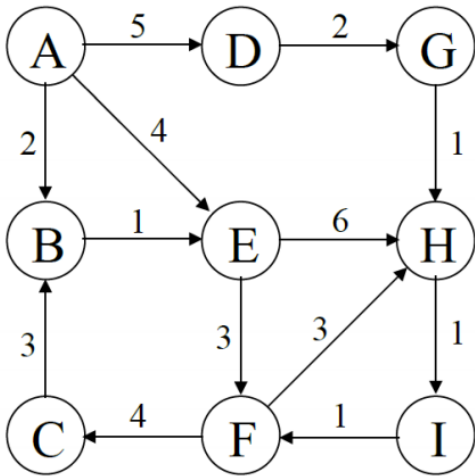


	toBeChecked	currDist	pred
A	t f	0	\
B	t f	∞ 2	/ A
C	t	∞ 10	/ F
D	t f	∞ 5	/ A
E	t f	∞ 4 3	/ A B
F	t f	∞ 6	/ E
G	t f	∞ 7	/ D
H	t f	∞ 9 8	/ E G
I	t f	∞ 9	/ H



v is vertex C

- u is empty



	toBeChecked	currDist	pred
A	t f	0	\
B	t f	∞ 2	/ A
C	t f	∞ 10	/ F
D	t f	∞ 5	/ A
E	t f	∞ 4 3	/ A B
F	t f	∞ 6	/ E
G	t f	∞ 7	/ D
H	t f	∞ 9 8	/ E G
I	t f	∞ 9	/ H

On next page, you will find another weighted directional graph for which you need to calculate the shortest distance.

We want to find the shortest path
from node 1 to all other nodes using
Dijkstra's algorithm.

