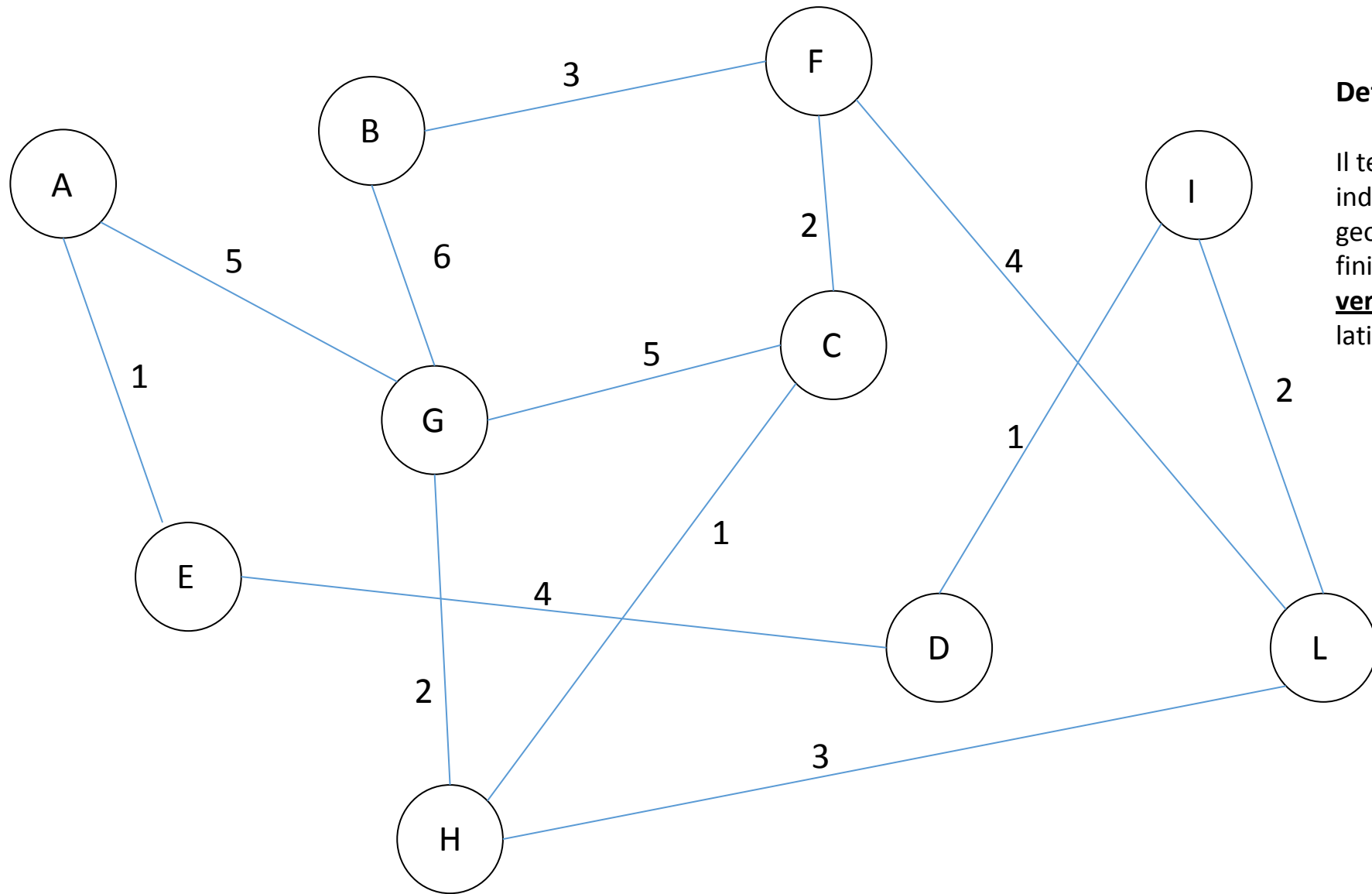


Algoritmo di Dijkstra

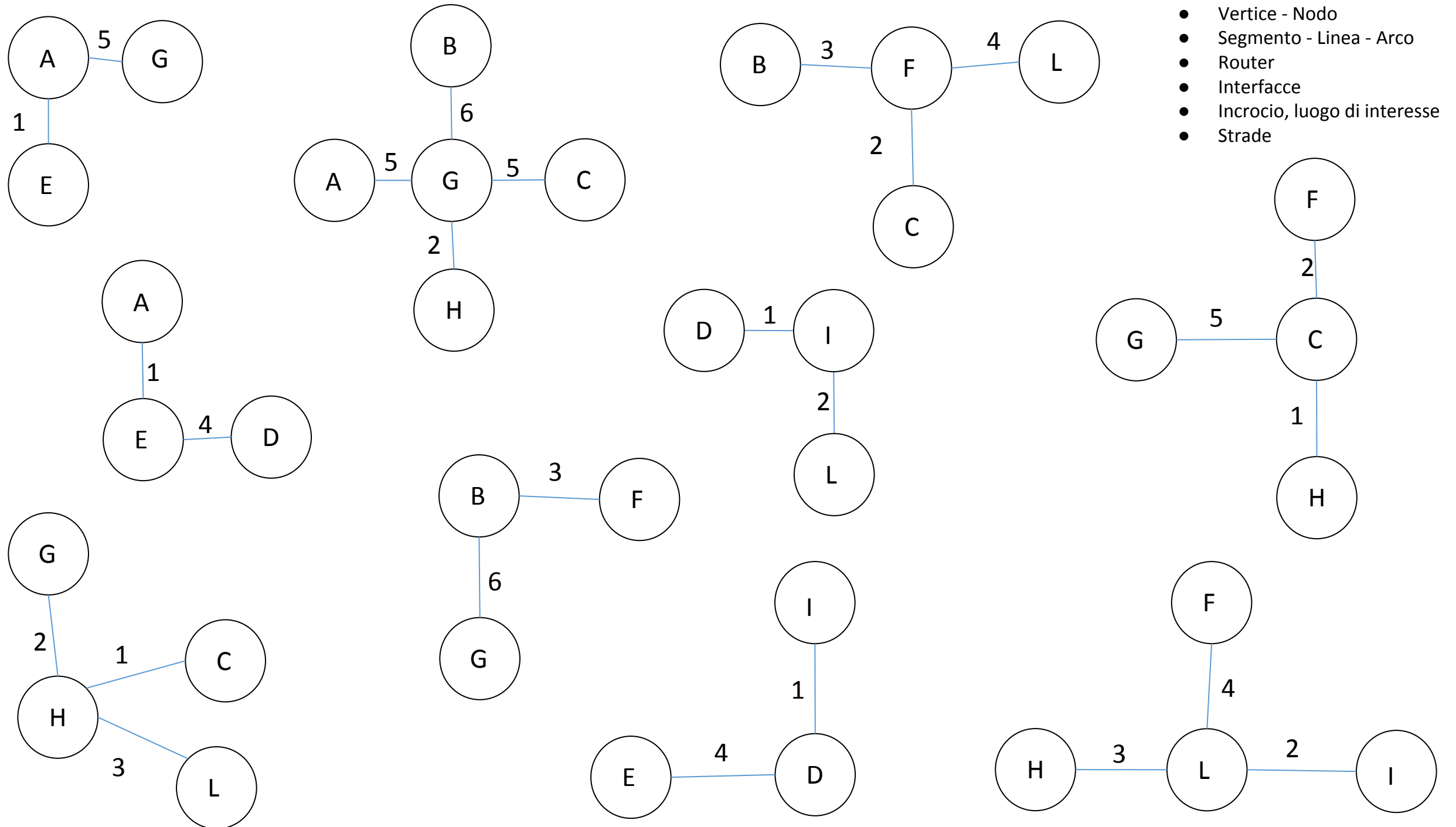
L'algoritmo di Dijkstra trova il cammino minimo tra un dato nodo (denominato "sorgente") e tutti gli altri nodi in un grafo.

Questo algoritmo utilizza i pesi degli archi per trovare il percorso che minimizza la distanza totale tra il nodo sorgente e gli altri nodi.

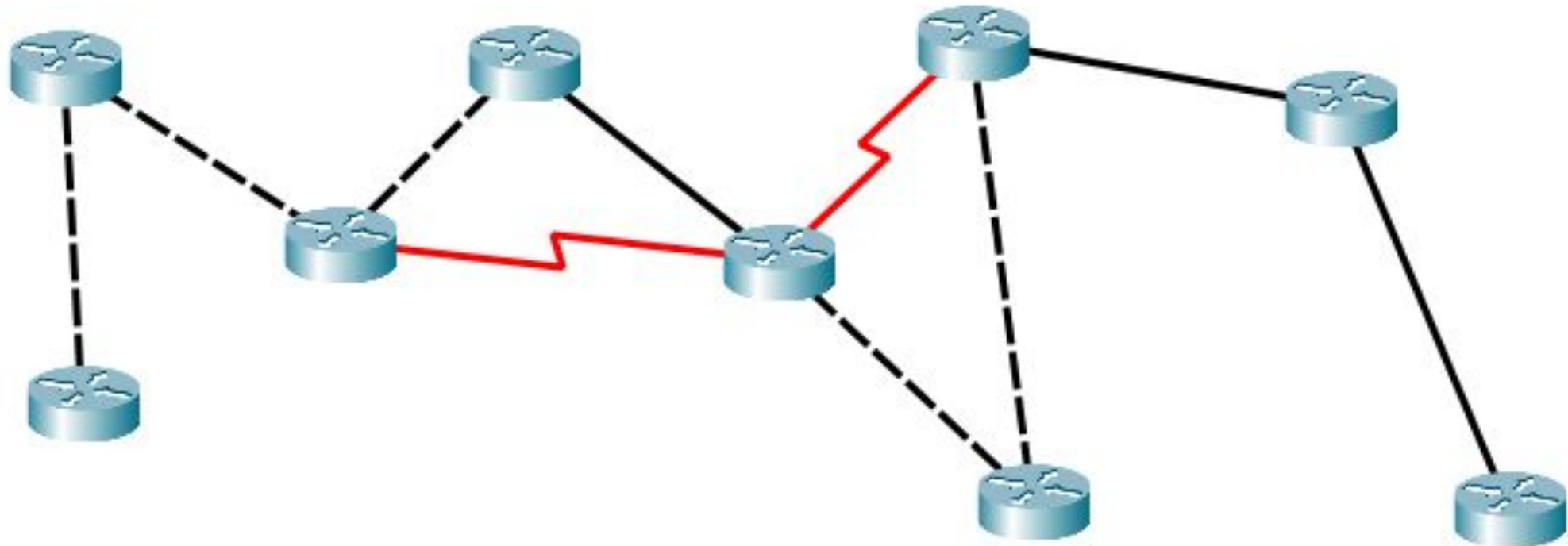


Definizione di grafo

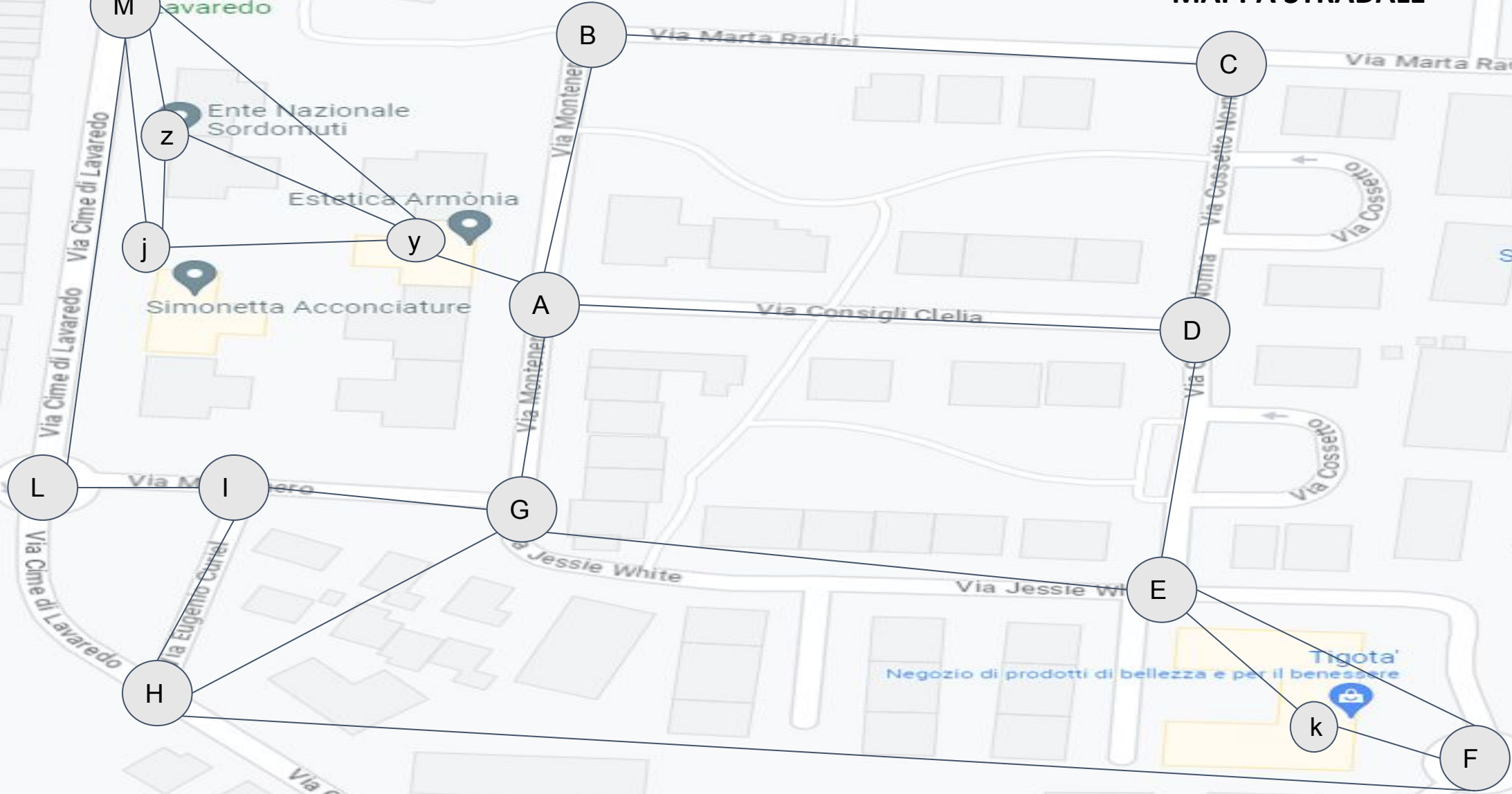
Il termine grafo viene usato per indicare una particolare figura geometrica, costituita da un insieme finito di punti, detti **nodi** (oppure **vertici**), e da segmenti o **archi**, detti lati, che congiungono **coppie di nodi**.

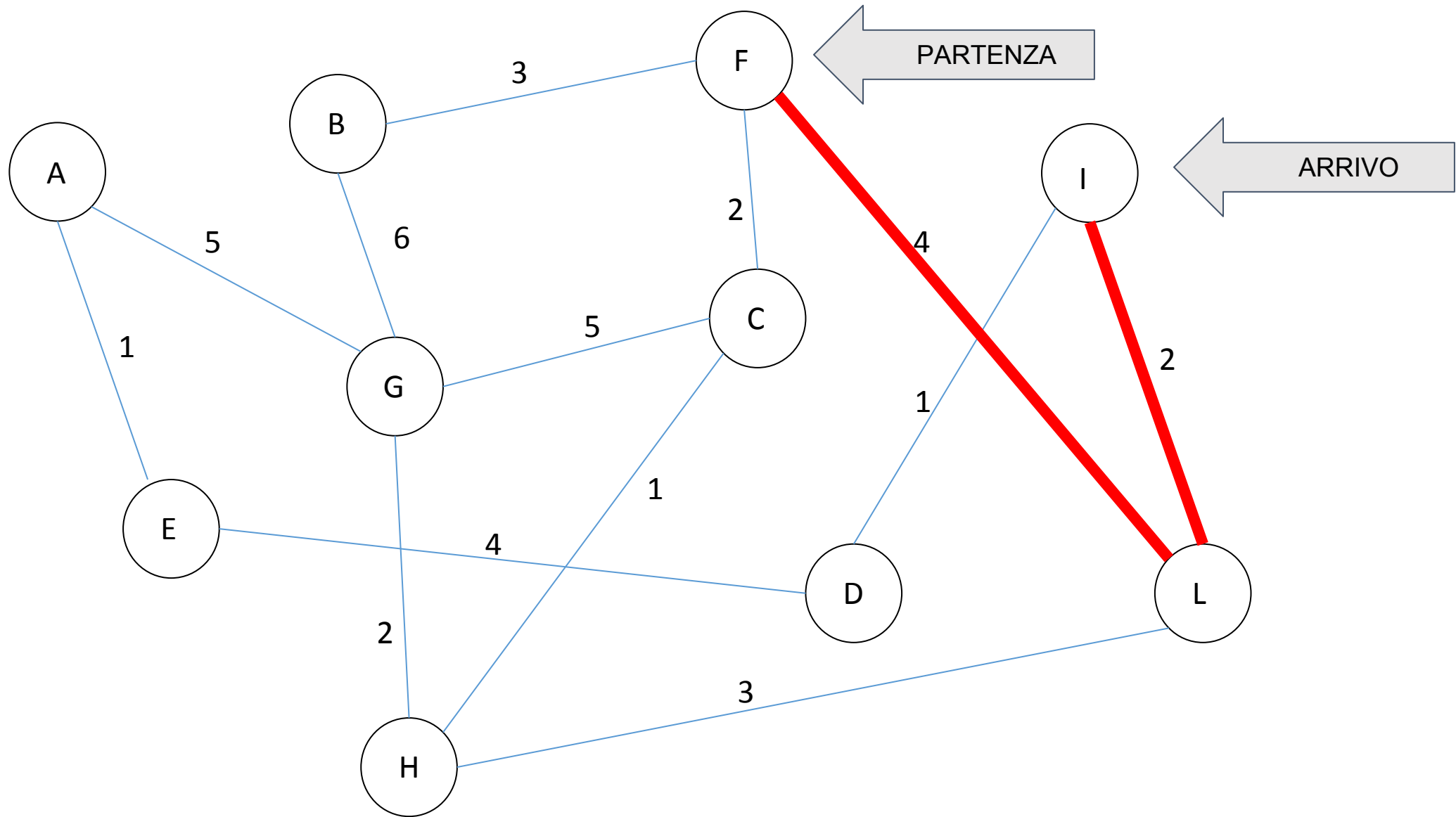


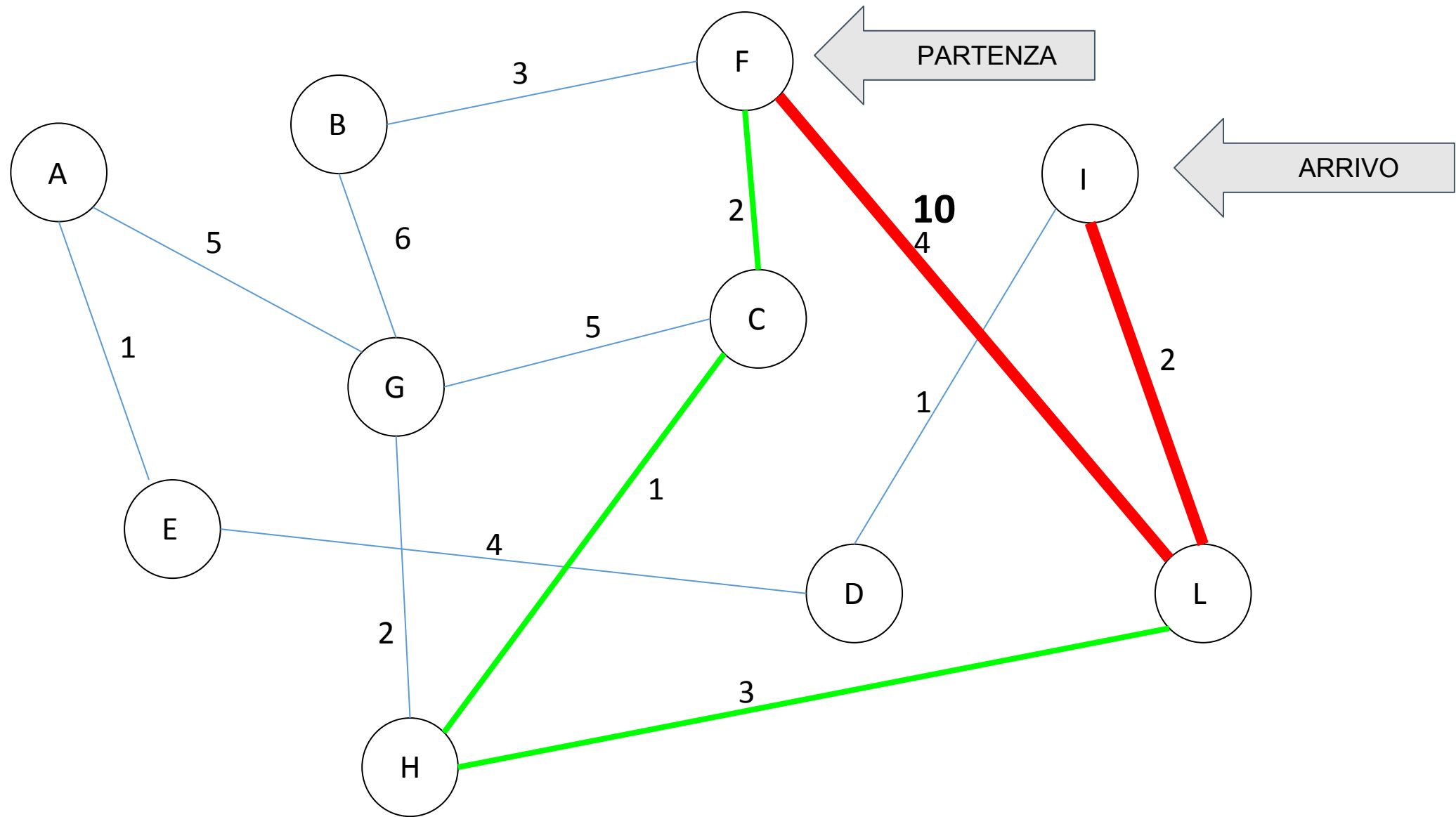
RETE INFORMATICA

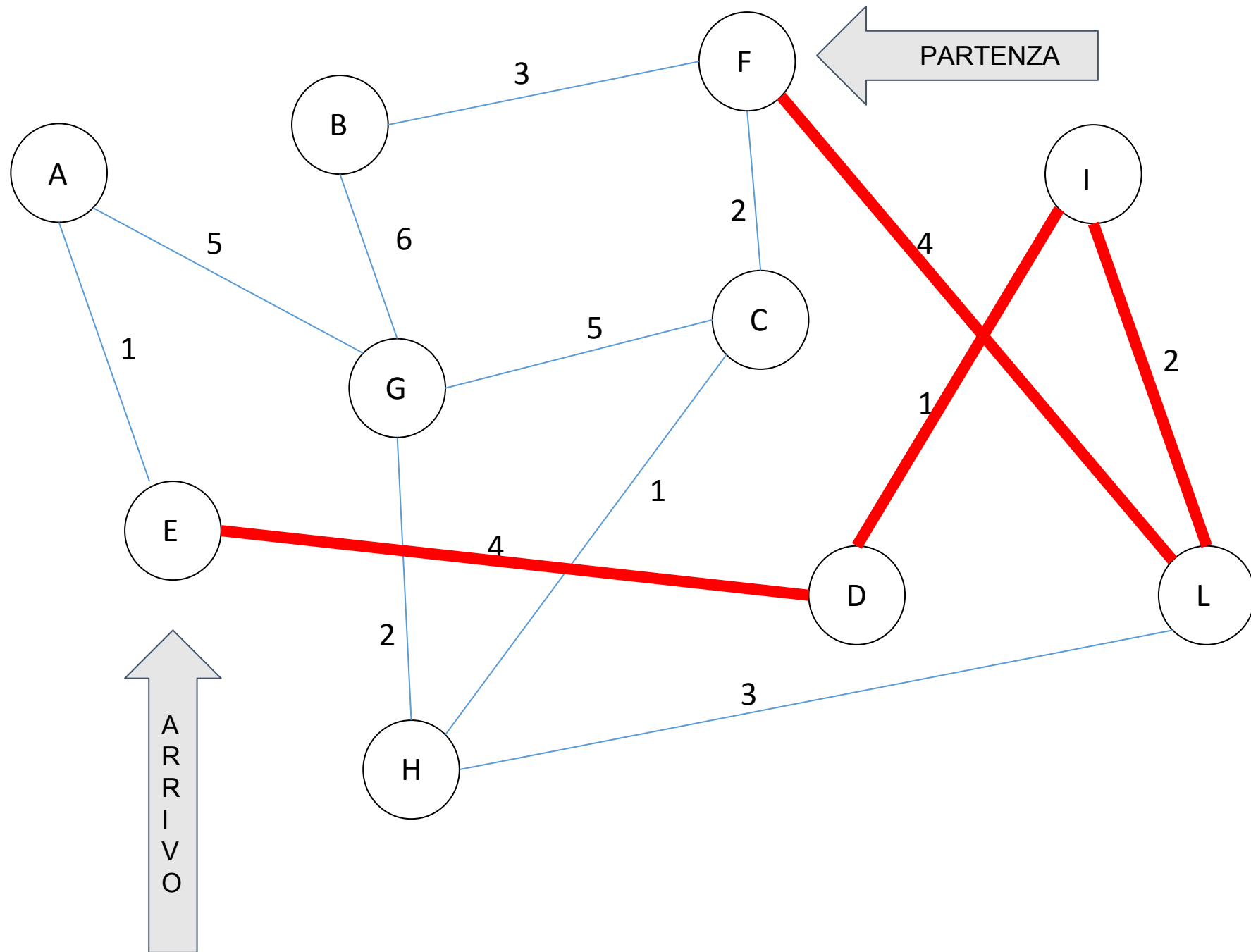


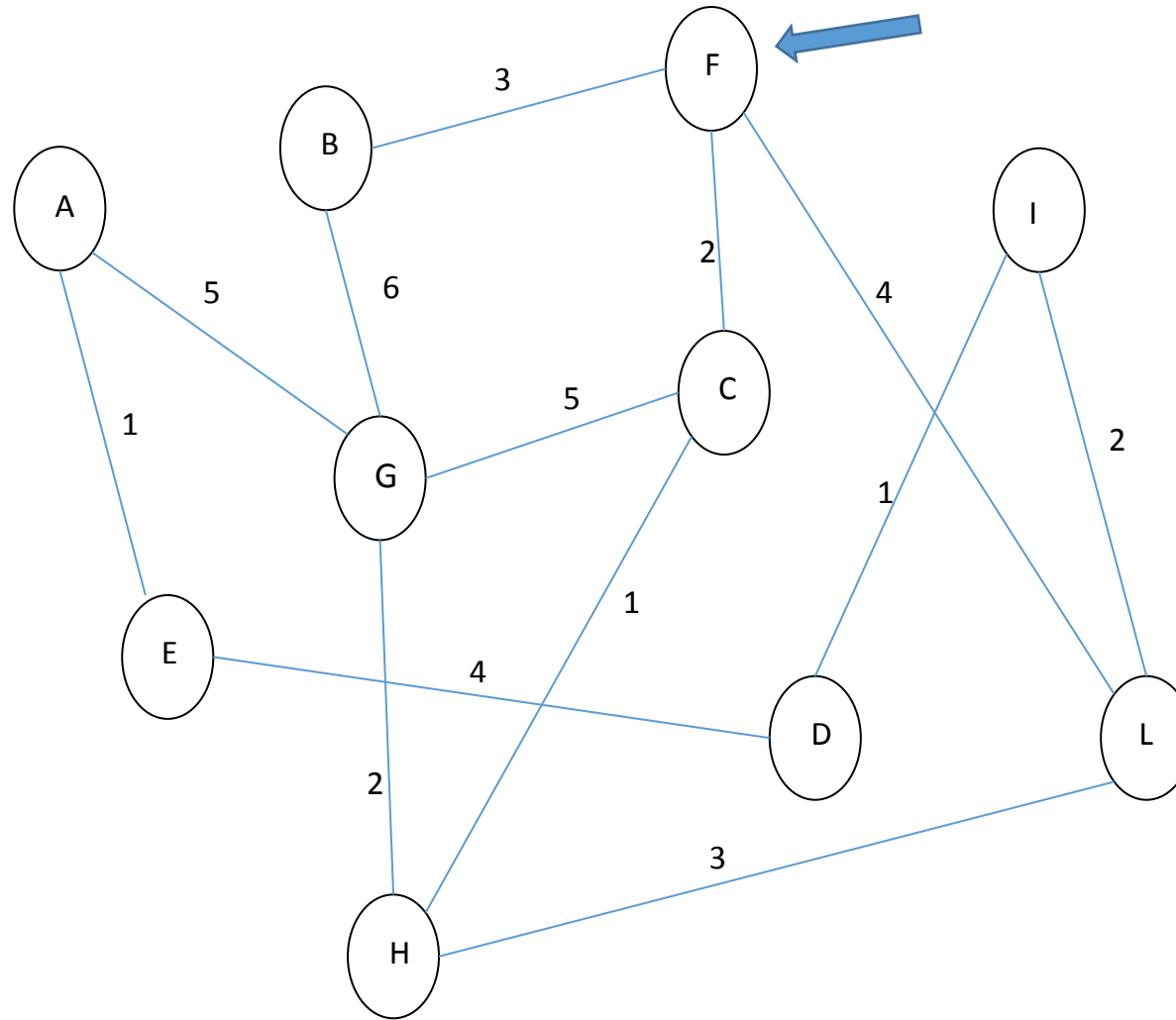
MAPPA STRADALE











Problema:

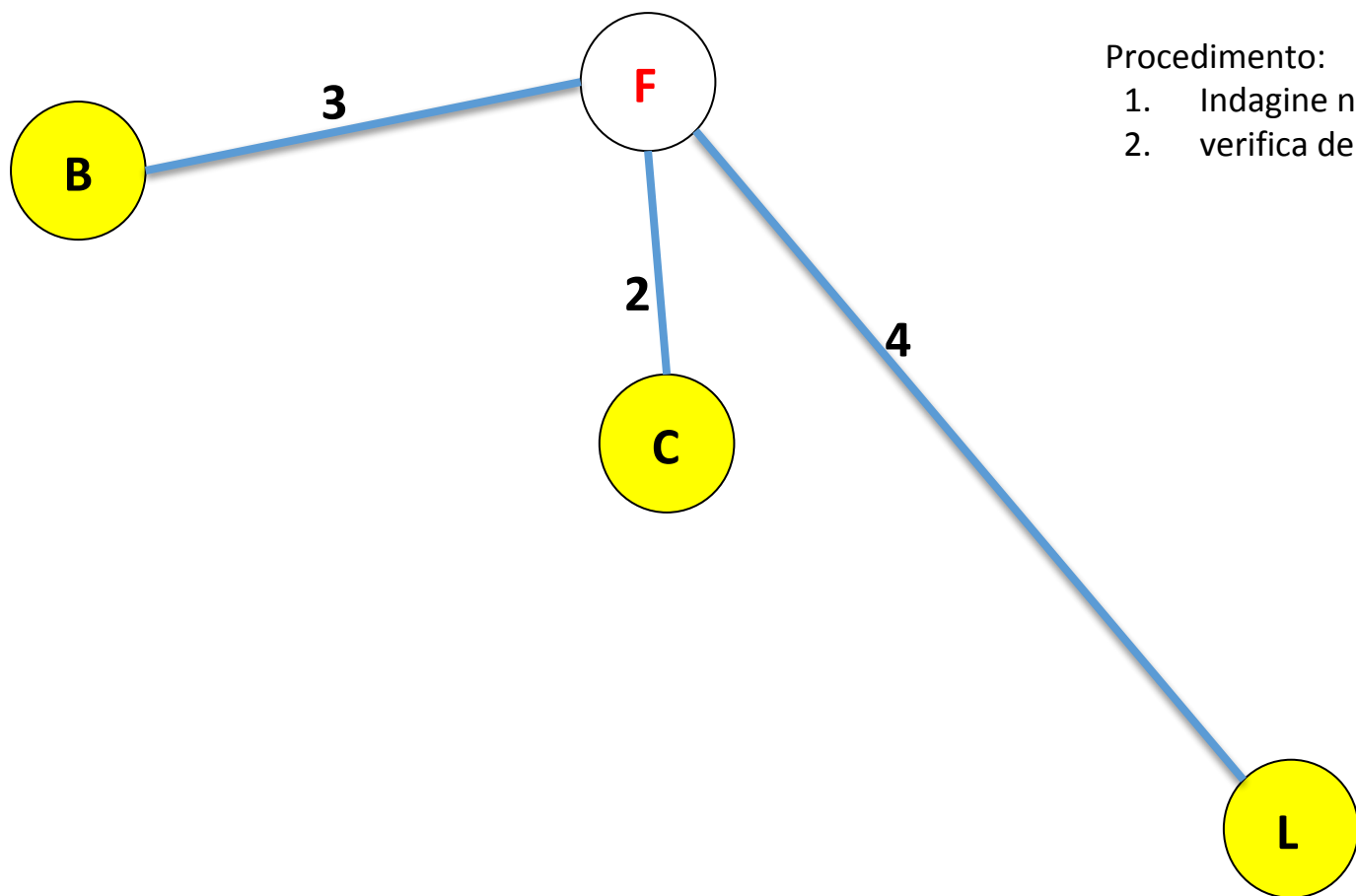
- Partenza: Vertice F

Obiettivo:

- Partendo da F trovare i cammini minimi verso gli altri vertici

Memorizziamo in una coda tutti i vertici del grafo. Il primo elemento del grafo è il vertice di partenza. Impostare i costi dei cammini ad infinito (non raggiungibili) per tutti i vertici del grafo e a 0 il costo del vertice iniziale a 0 (da F a F costo 0).

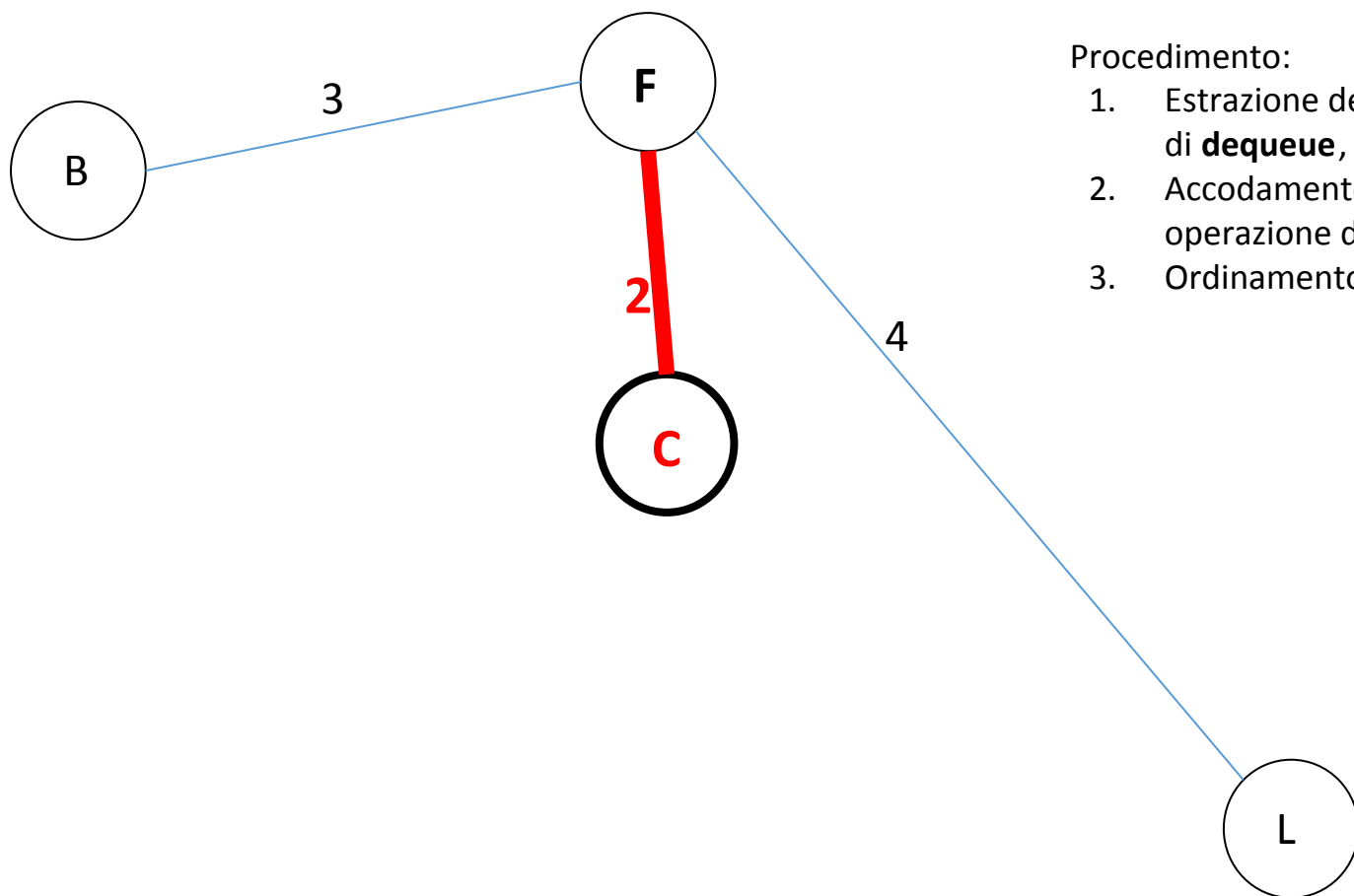
Vertici	Costo	DA
F	0	
A	∞	
B	∞	
C	∞	
D	∞	
E	∞	
G	∞	
H	∞	
I	∞	
L	∞	



Procedimento:

1. Indagine nodi adiacenti
2. verifica del cammino minimo

Vertici	Costo	DA
F	0	
A	∞	
B	3	F
C	2	F
D	∞	
E	∞	
G	∞	
H	∞	
I	∞	
L	4	F

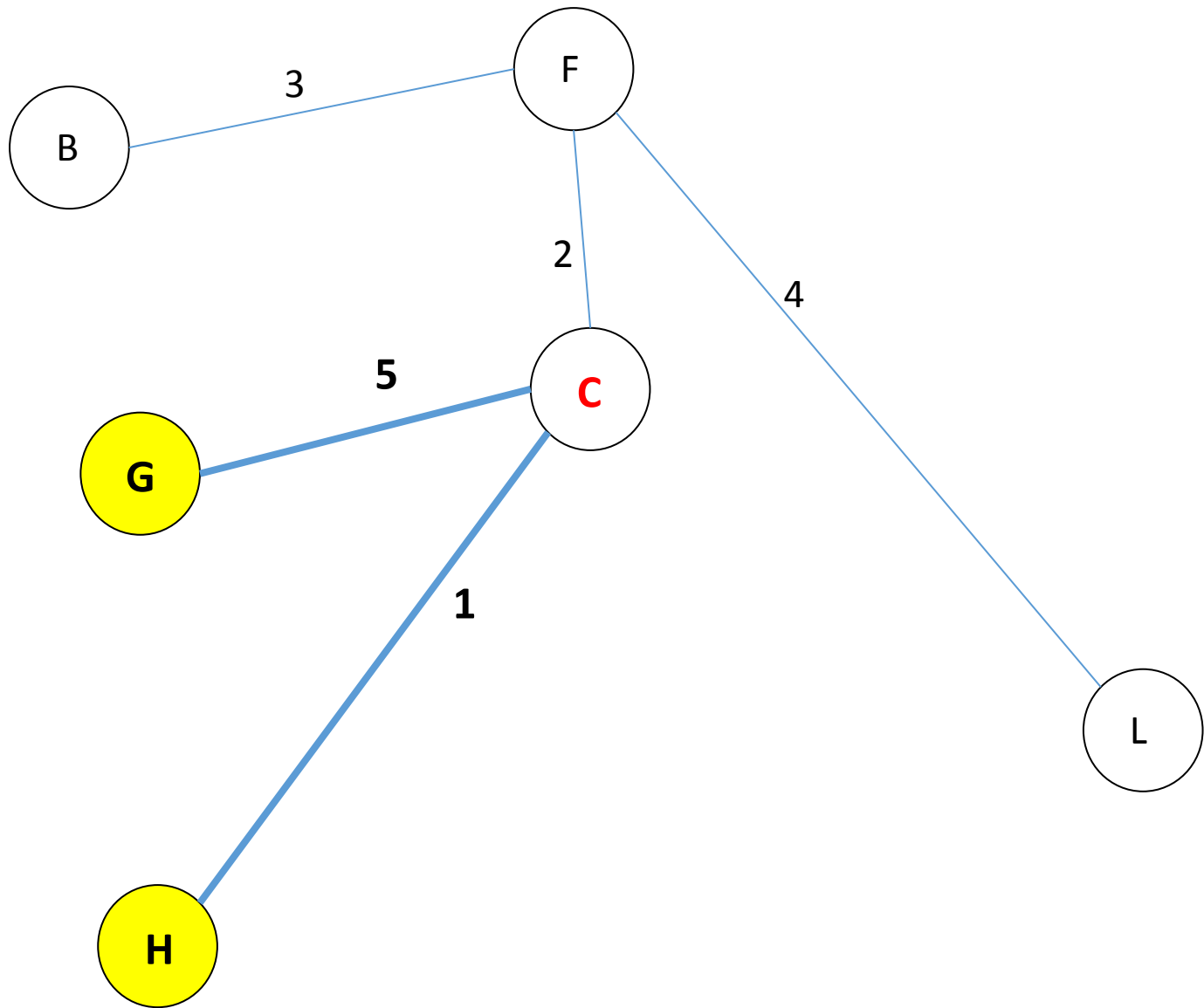


Procedimento:

1. Estrazione del nodo F dalla coda di indagine - Detta anche operazione di **dequeue**, serve a rimuovere un elemento dalla testa della coda.
2. Accodamento del nodo F nella coda cammini minimi - Detta anche operazione di **enqueue**, serve a mettere un elemento in coda.
3. Ordinamento degli elementi in base al costo

Vertici	Costo	DA
F	0	

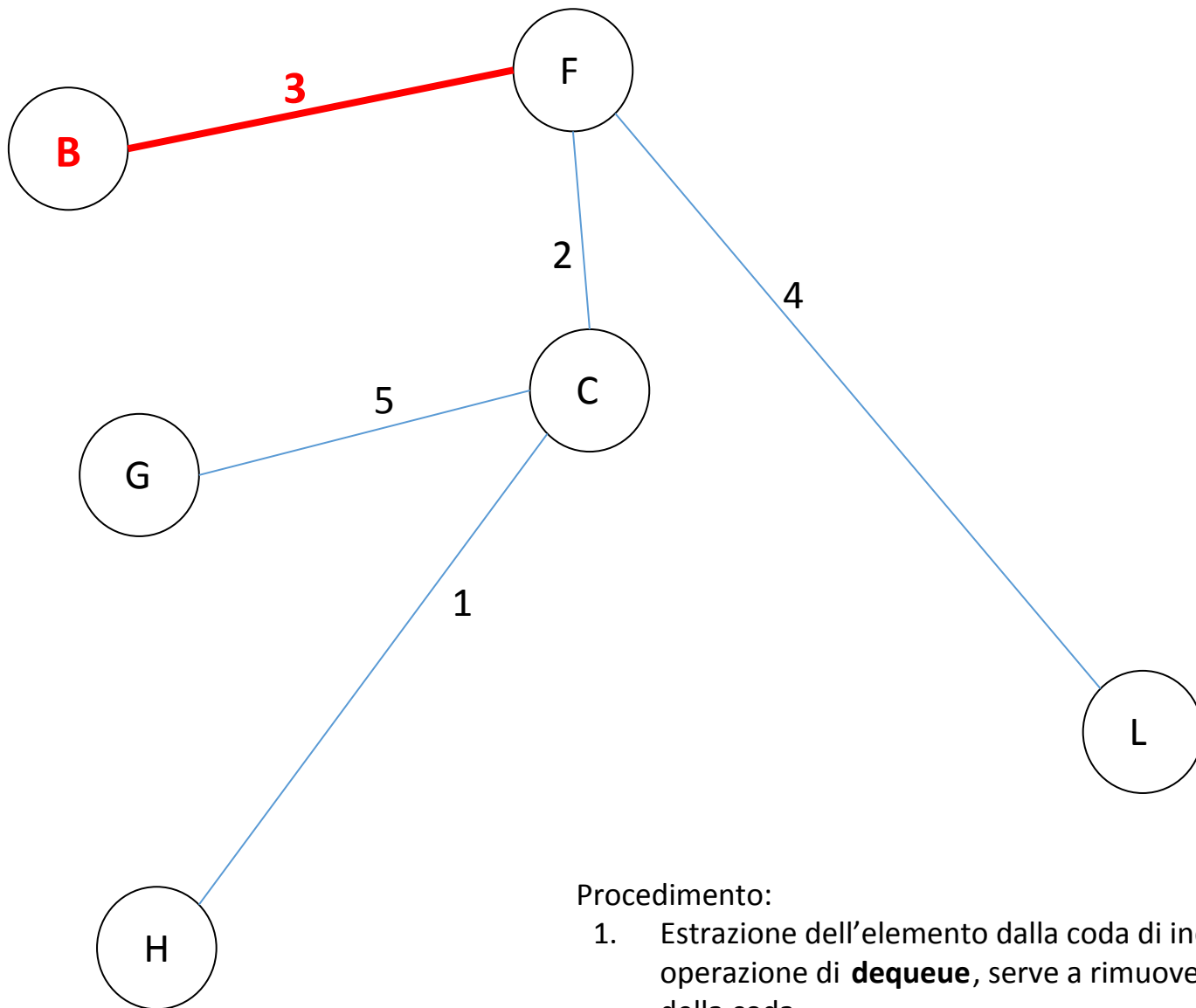
Vertici	Costo	DA
C	2	F
B	3	F
L	4	F
A	∞	
D	∞	
E	∞	
G	∞	
H	∞	
I	∞	



- Procedimento:
1. Indagine nodi adiacenti
 2. verifica del cammino minimo

Vertici	Costo	DA
F	0	

Vertici	Costo	DA
C	2	F
B	3	F
L	4	F
A	∞	
D	∞	
E	∞	
G	2+5	C
H	2+1	C
I	∞	

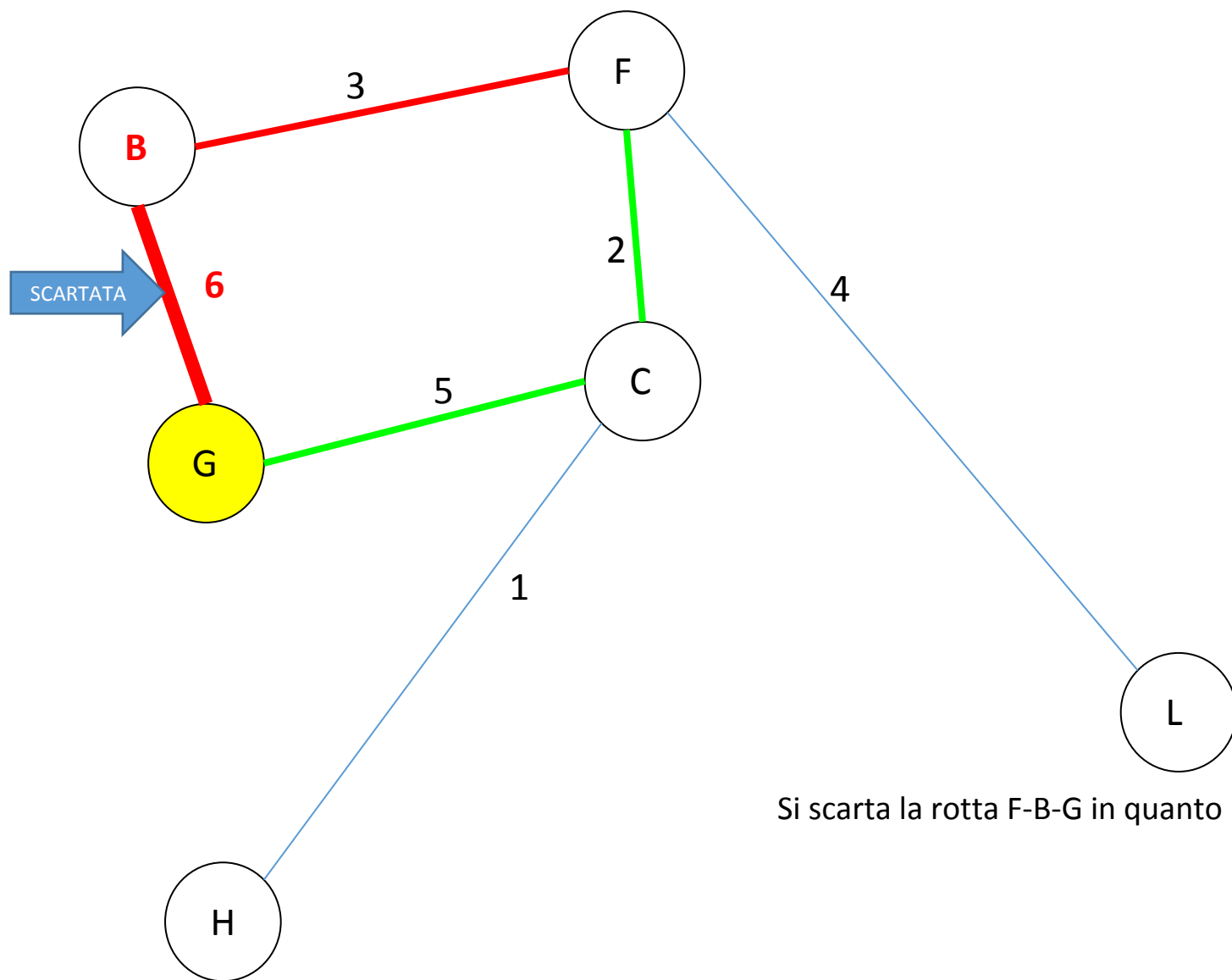


Vertici	Costo	DA
F	0	
C	2	F

Vertici	Costo	DA
B	3	F
H	3	C
L	4	F
G	7	C
A	∞	
D	∞	
E	∞	
I	∞	

Procedimento:

1. Estrazione dell'elemento dalla coda di indagine - Detta anche operazione di **dequeue**, serve a rimuovere un elemento dalla testa della coda.
2. Accodamento dell'elemento estratto nella coda cammini minimi - Detta anche operazione di **enqueue**, serve a mettere un elemento in coda.
3. Ordinamento degli elementi in base al costo



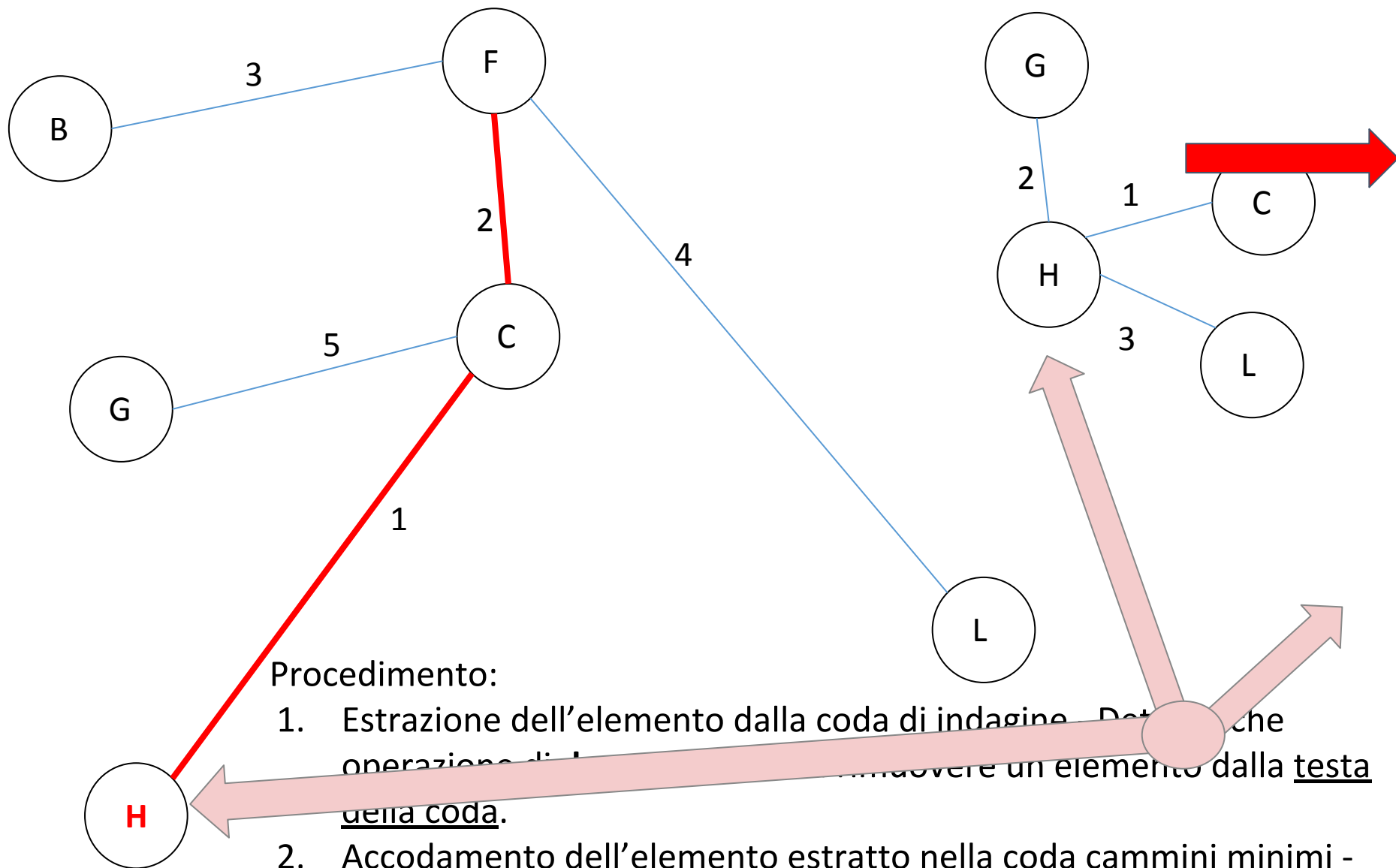
Si scarta la rotta F-B-G in quanto più costosa rispetto a F-C-G

Procedimento:

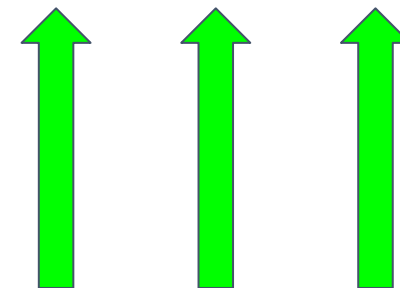
1. Indagine nodi adiacenti
2. verifica del cammino minimo

Vertici	Costo	DA
F	0	
C	2	F

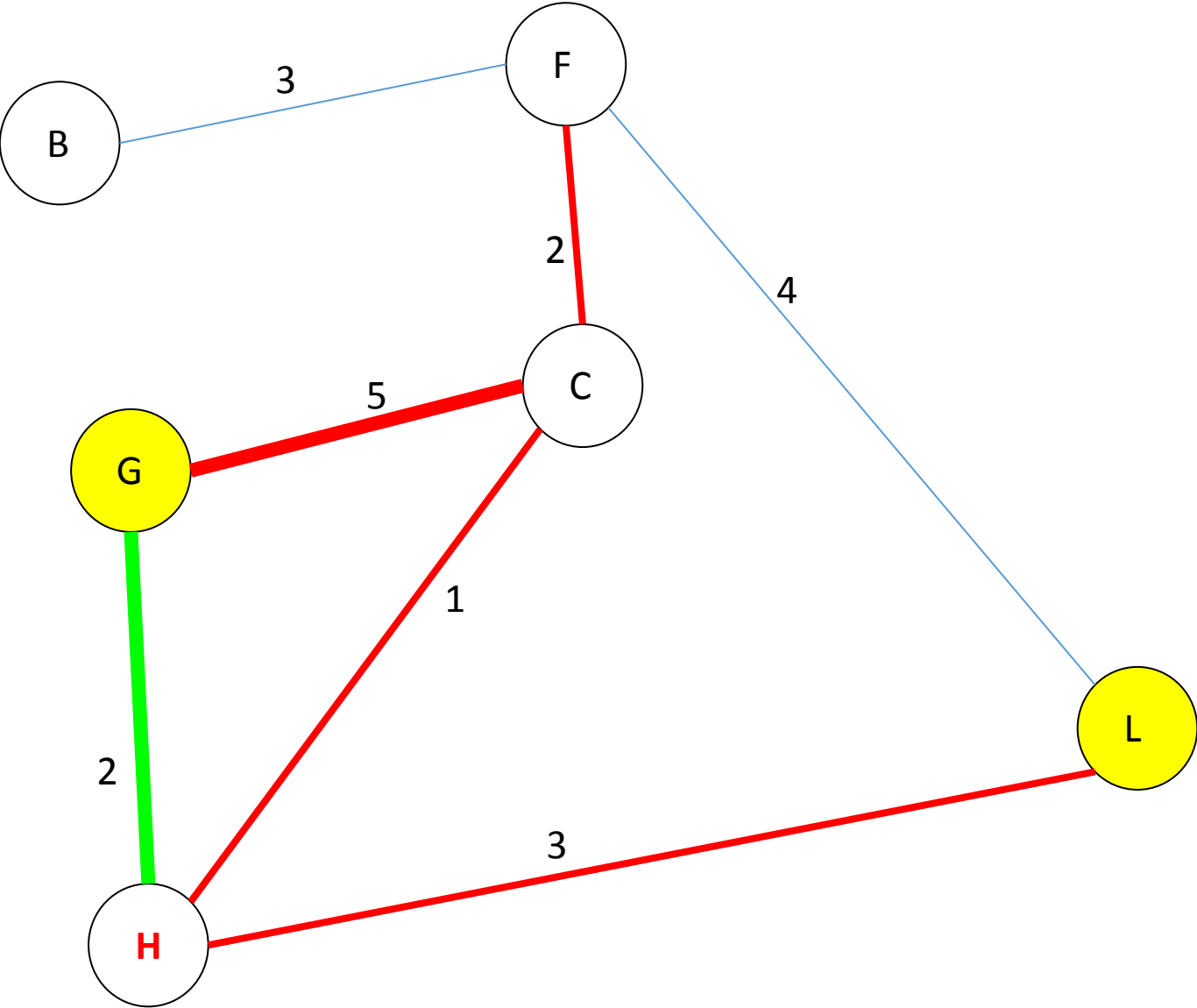
Vertici	Costo	DA
B	3	F
H	3	C
L	4	F
G	7 (3+6)	C B
A	∞	
D	∞	
E	∞	
I	∞	



Vertici	Costo	DA
F	0	
C	2	F
B	3	F



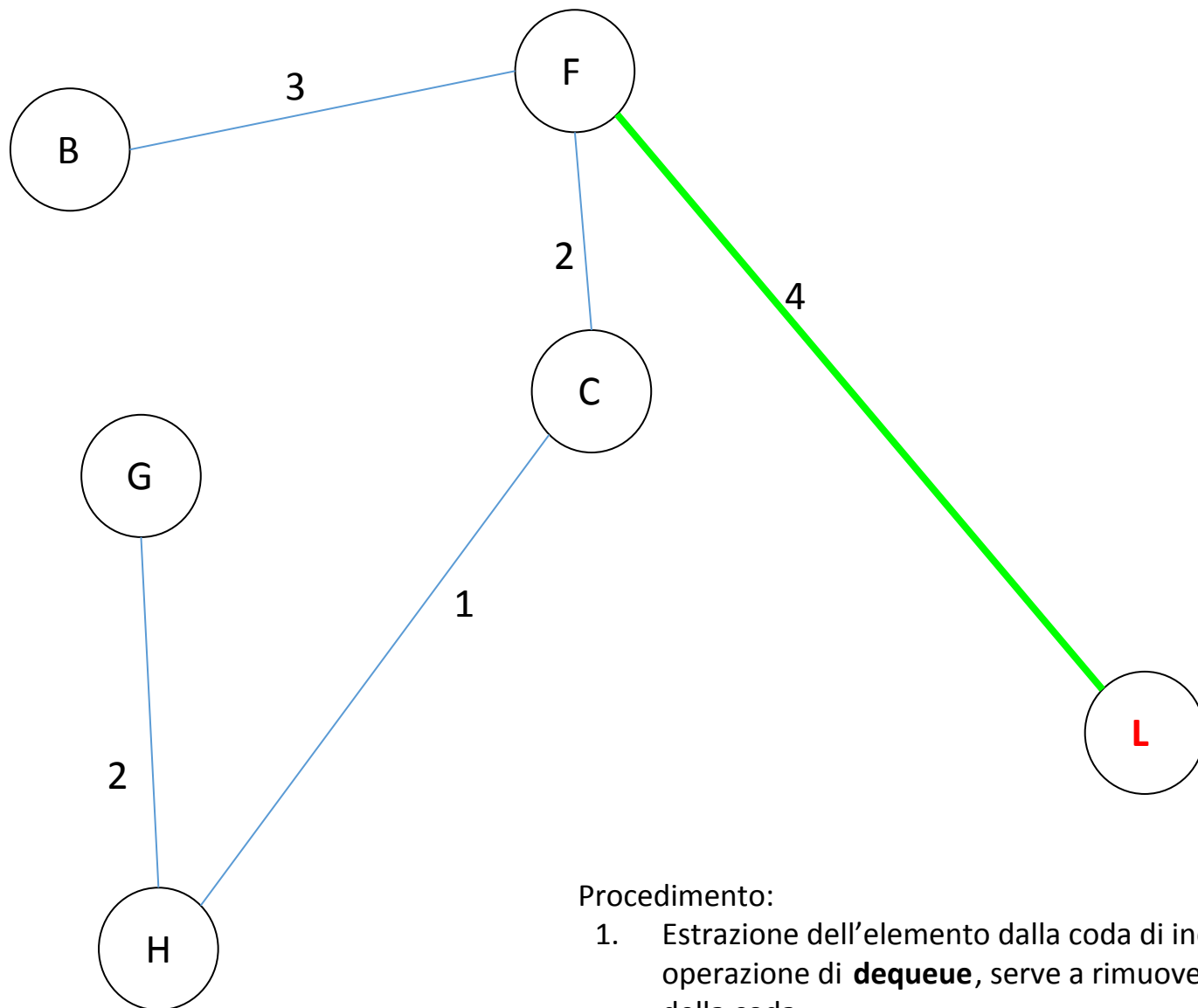
Vertici	Costo	DA
H	3	C
L	4	F
G	7	C
A	∞	
D	∞	
E	∞	
I	∞	



- Procedimento:
- 1. Indagine nodi adiacenti
 - 2. verifica del cammino minimo

Vertici	Costo	DA
F	0	
C	2	F
B	3	F

Vertici	Costo	DA
H	3	C
L	4	F
3+3		
G	7	C
3+2		
A	∞	
D	∞	
E	∞	
I	∞	

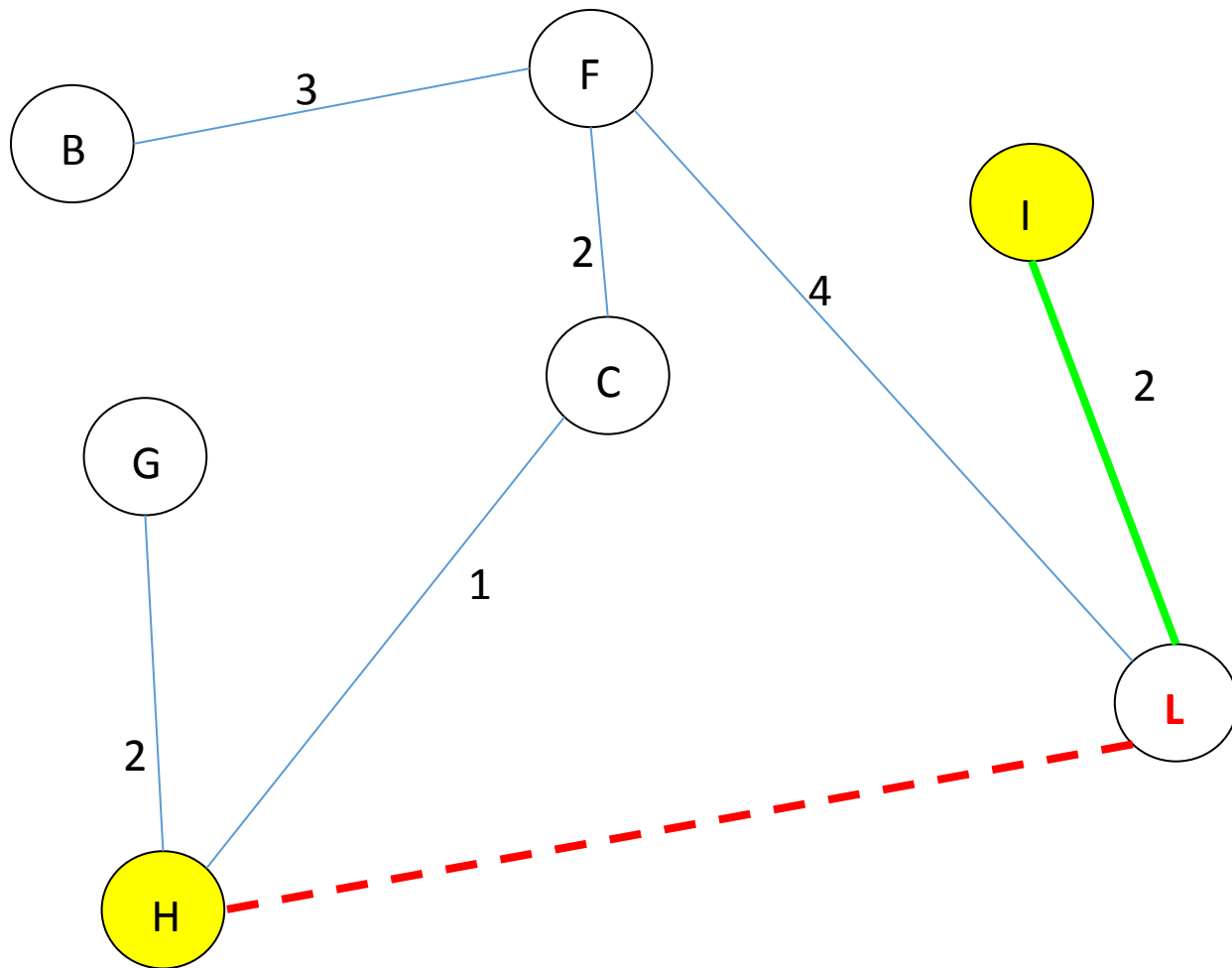


Vertici	Costo	DA
F	0	
C	2	F
B	3	F
H	3	C

Vertici	Costo	DA
L	4	F
G	5	H
A	∞	
D	∞	
E	∞	
I	∞	

Procedimento:

1. Estrazione dell'elemento dalla coda di indagine - Detta anche operazione di **dequeue**, serve a rimuovere un elemento dalla testa della coda.
2. Accodamento dell'elemento estratto nella coda cammini minimi - Detta anche operazione di **enqueue**, serve a mettere un elemento in coda.
3. Ordinamento degli elementi in base al costo



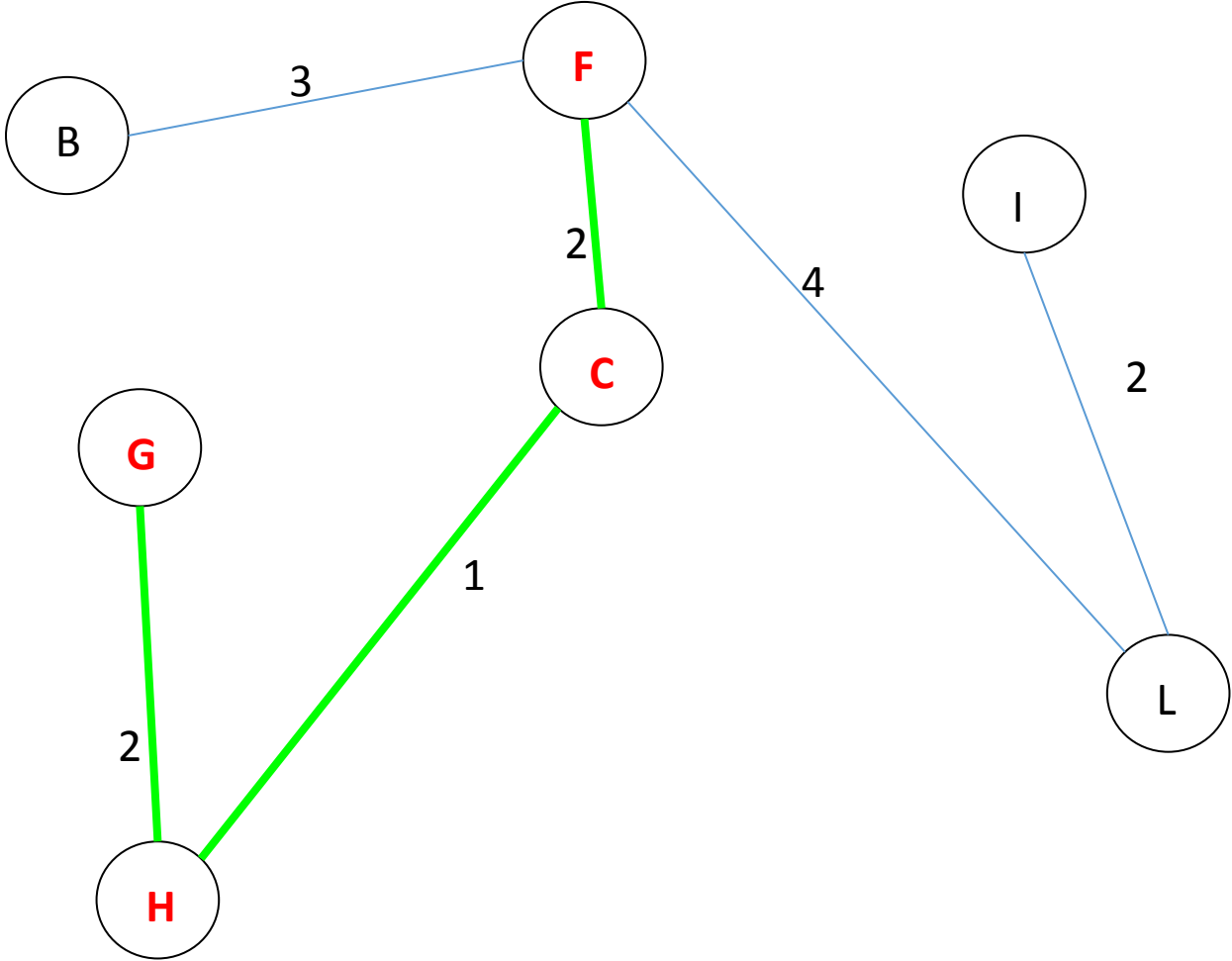
Vertici	Costo	DA
F	0	
C	2	F
B	3	F
H	3	C

Già visitato

Vertic i	Costo	DA
L	4	F
G	5	H
A	∞	
D	∞	
E	∞	
I	$4+2=6$	L

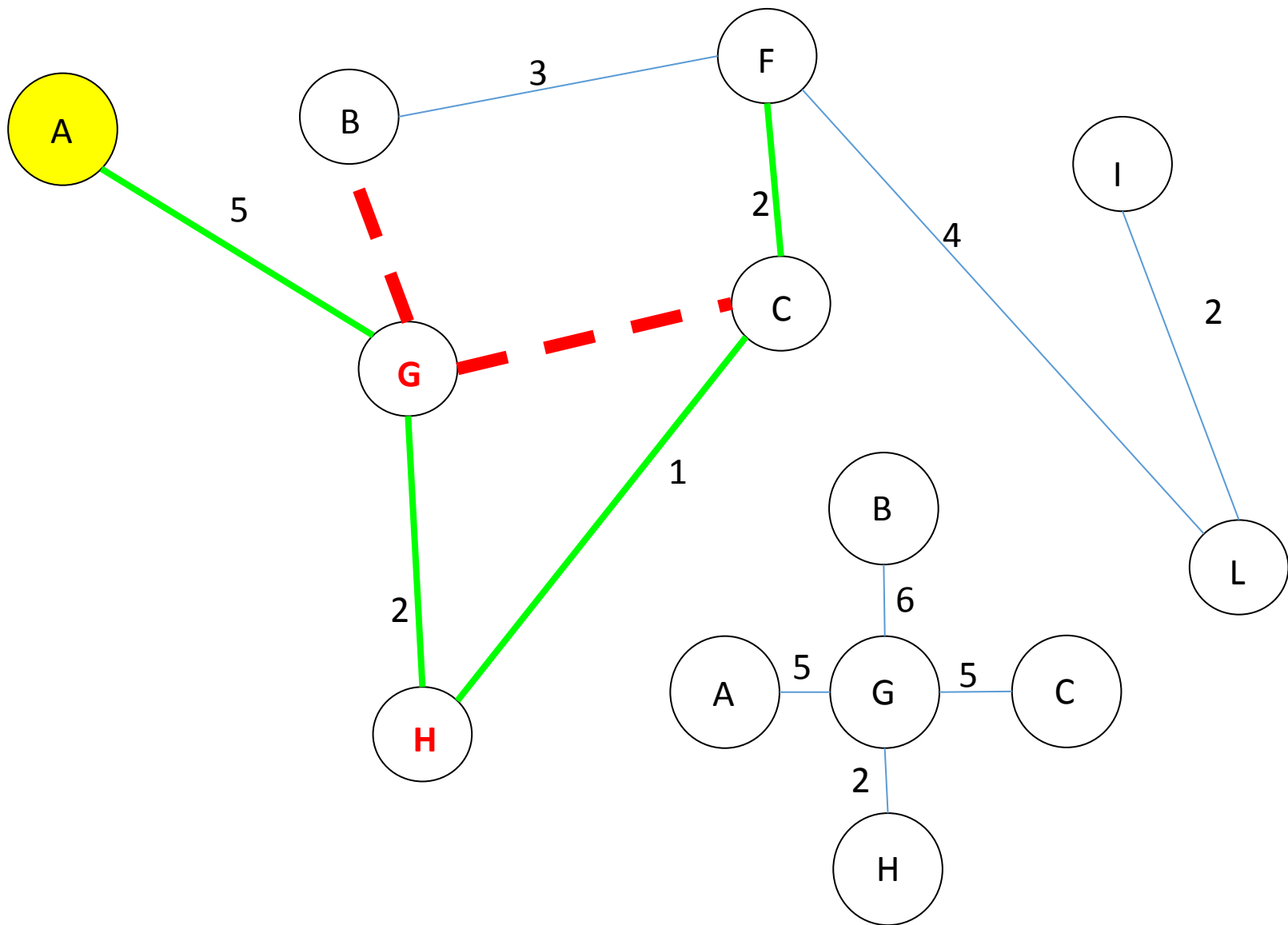
Cammino minimo da F a G

- Vertice F - Vertice C - Costo 2
- Vertice C - Vertice H - Costo 1
- Vertice H - Vertice G - Costo 2
- Totale costo 5



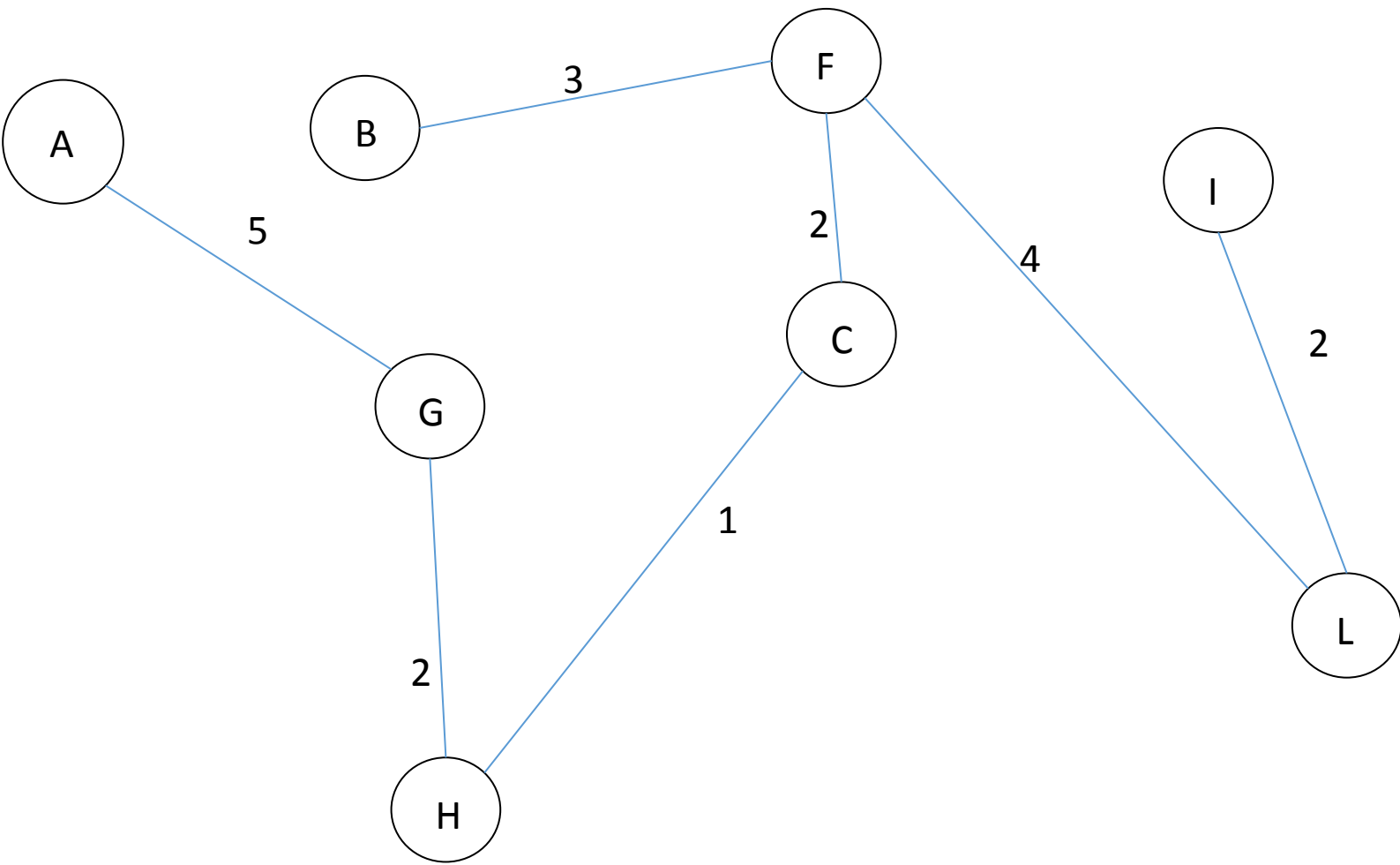
Vertici	Costo	DA
F	0	
C	2	F
B	3	F
H	3	C
L	4	F

Vertici	Costo	DA
G	5	H
I	6	L
A	∞	
D	∞	
E	∞	



Vertici	Costo	DA
F	0	
C	2	F
B	3	F
H	3	C
L	4	F

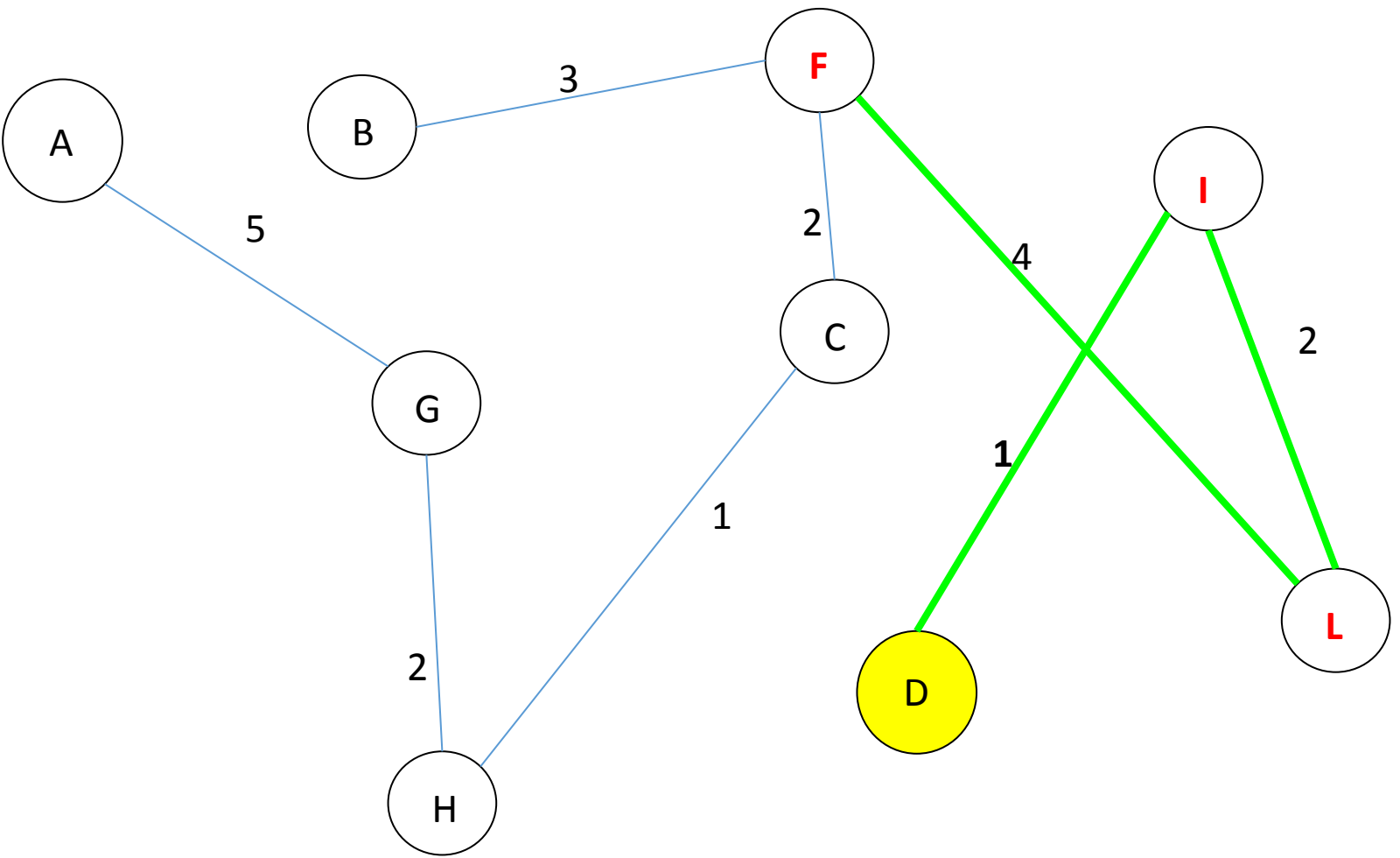
Vertici	Costo	DA
G	5	H
I	6	L
A	5+5=10	
D	∞	
E	∞	



Vertici	Costo	DA
F	0	
C	2	F
B	3	F
H	3	C
L	4	F

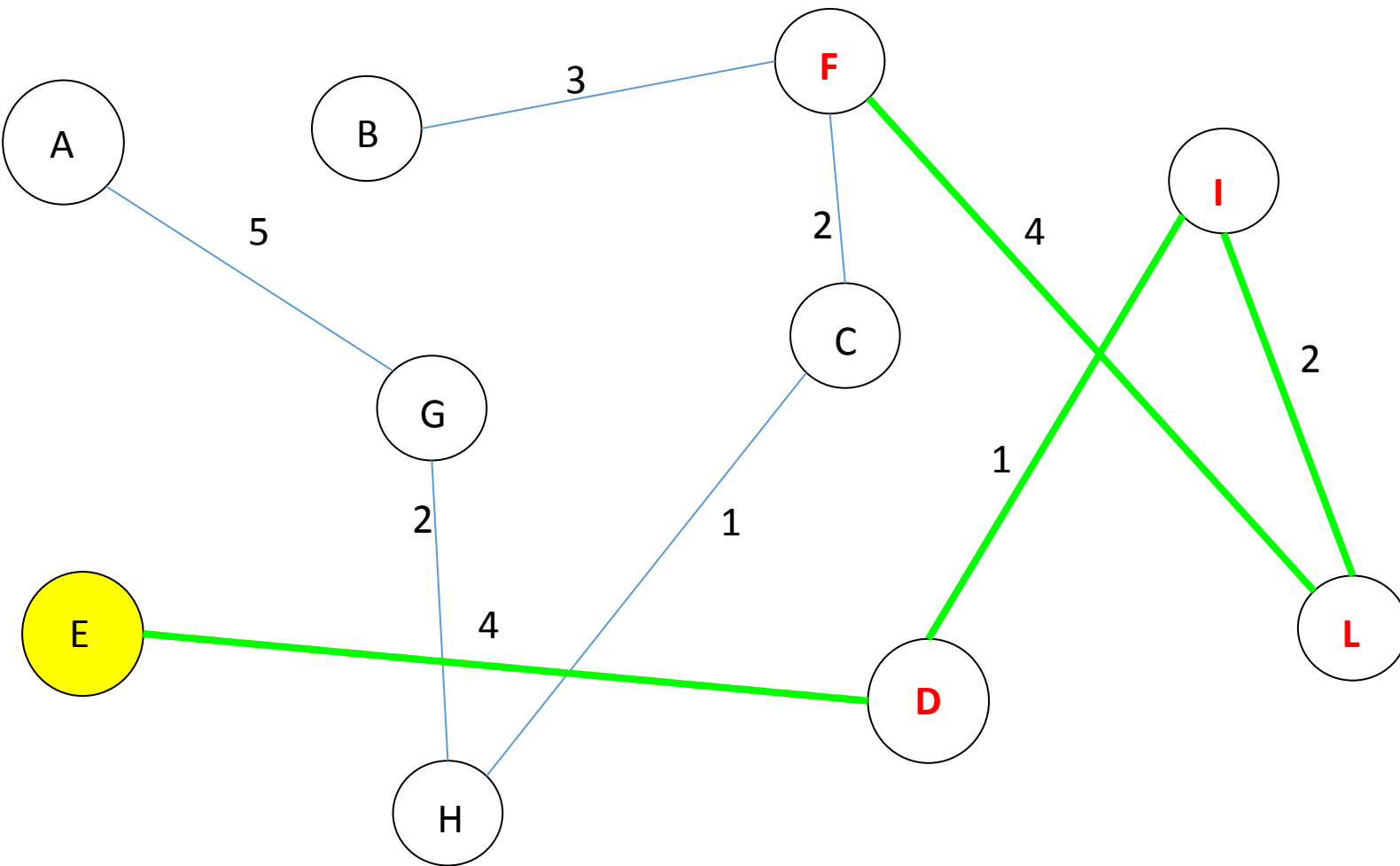
G S H

Vertici	Costo	DA
G	5	H
I	6	L
A	10	G
D	∞	
E	∞	



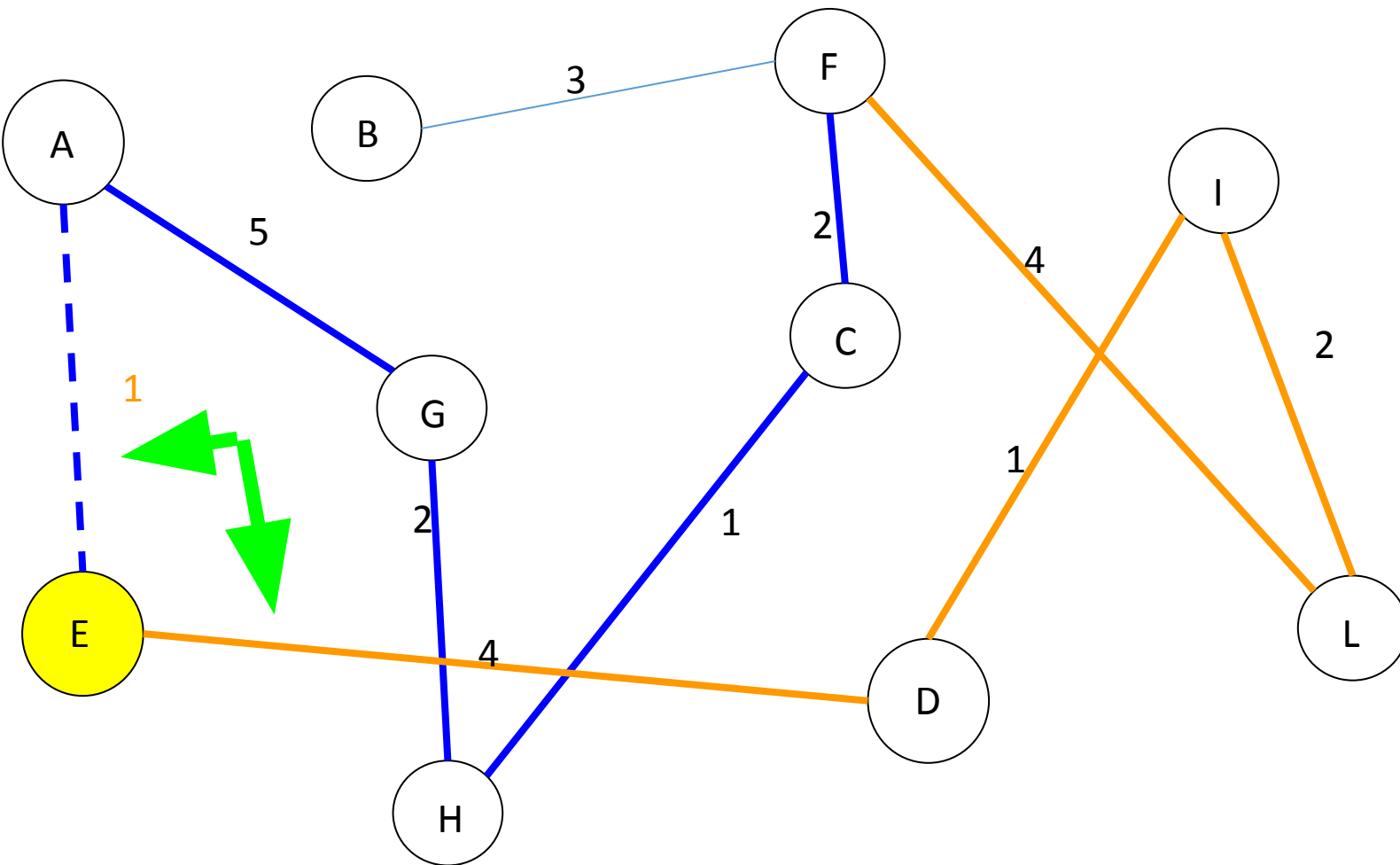
Vertici	Costo	DA
F	0	
C	2	F
B	3	F
H	3	C
L	4	F
G	5	H

Vertici	Costo	DA
I	6	L
A	10	G
D	6+1=7	I
E	∞	



Vertici	Costo	DA
F	0	
C	2	F
B	3	F
H	3	C
L	4	F
G	5	H
I	6	L

Vertici	Costo	DA
D	7	I
A	10	G
E	7+4 = 11	D



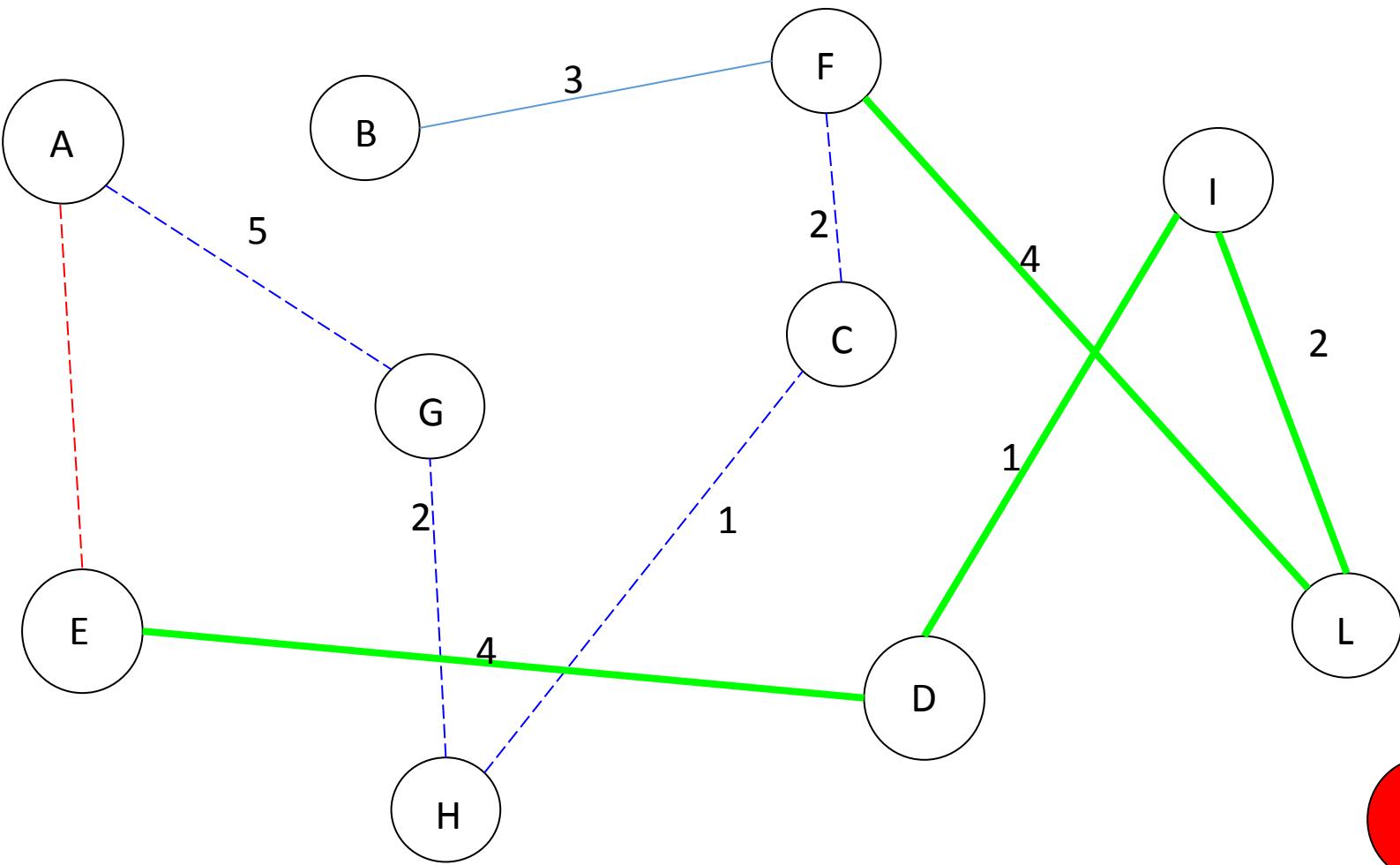
In questo particolare caso si individuano due casi di cammini minimi

- E - D - I - L - F
- E - A - G - H - C - I

La scelta dipende dall'algoritmo

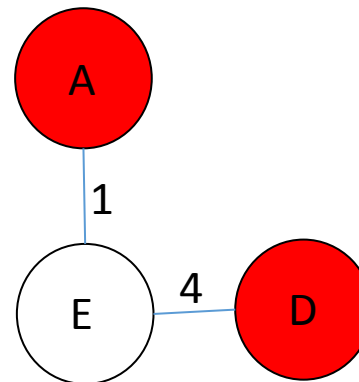
Vertici	Costo	DA
F	0	
C	2	F
B	3	F
H	3	C
L	4	F
G	5	H
I	6	L
D	7	I

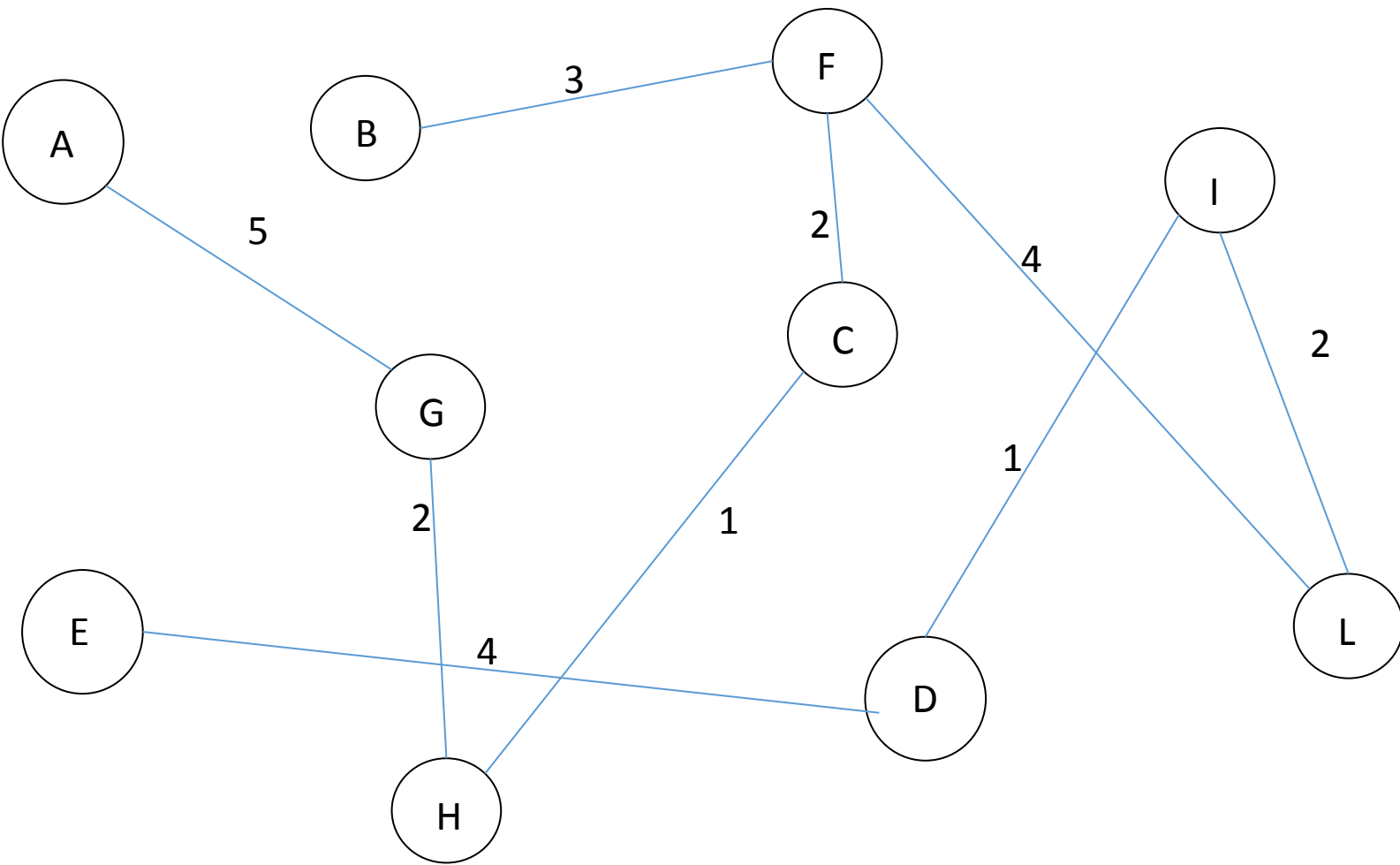
Vertici	Costo	DA
A	10	G
E	11 10+1= 11	D A



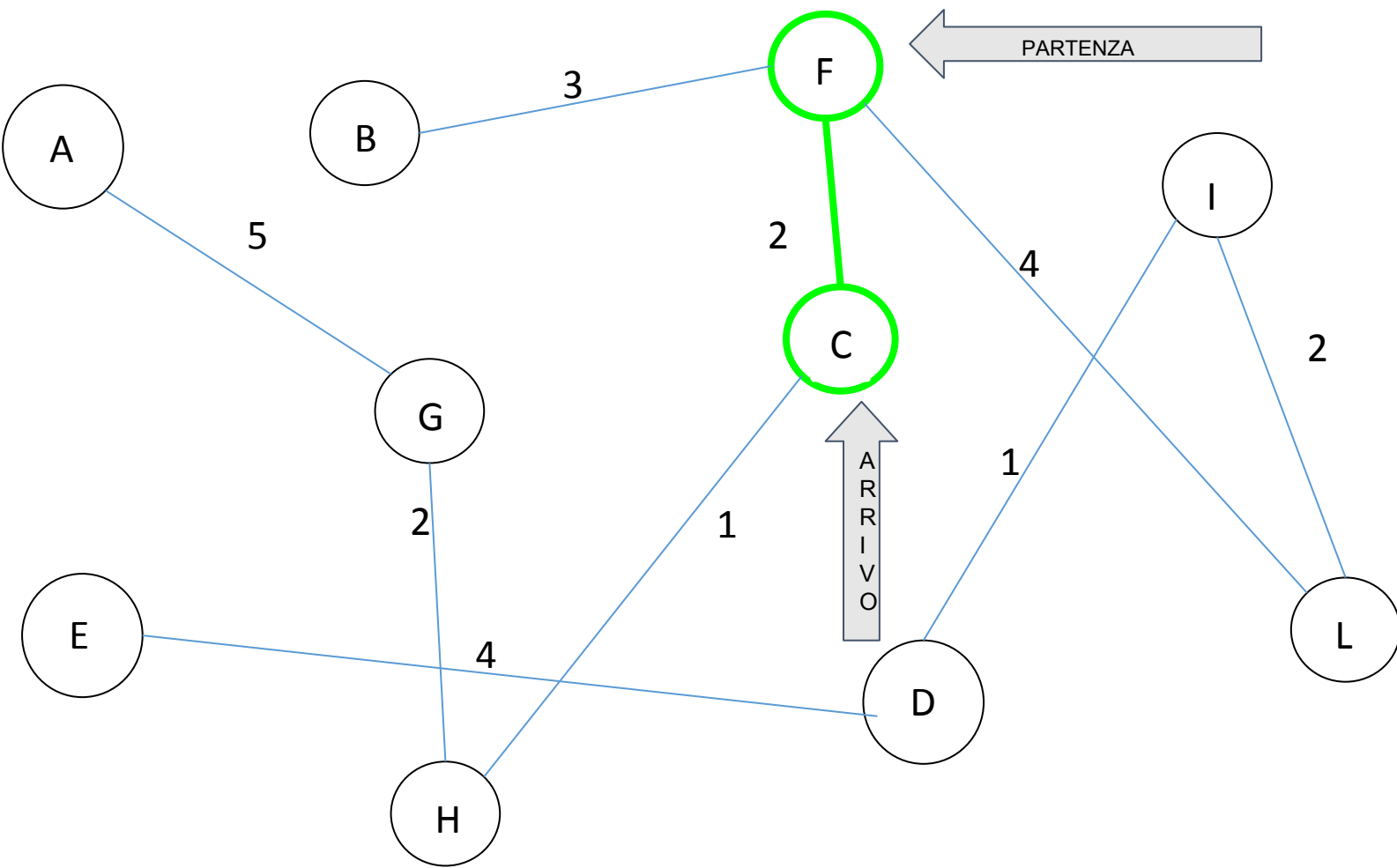
Vertici	Costo	DA
F	0	
C	2	F
B	3	F
H	3	C
L	4	F
G	5	H
I	6	L
D	7	I
A	10	G

Vertici	Costo	DA
E	11	D

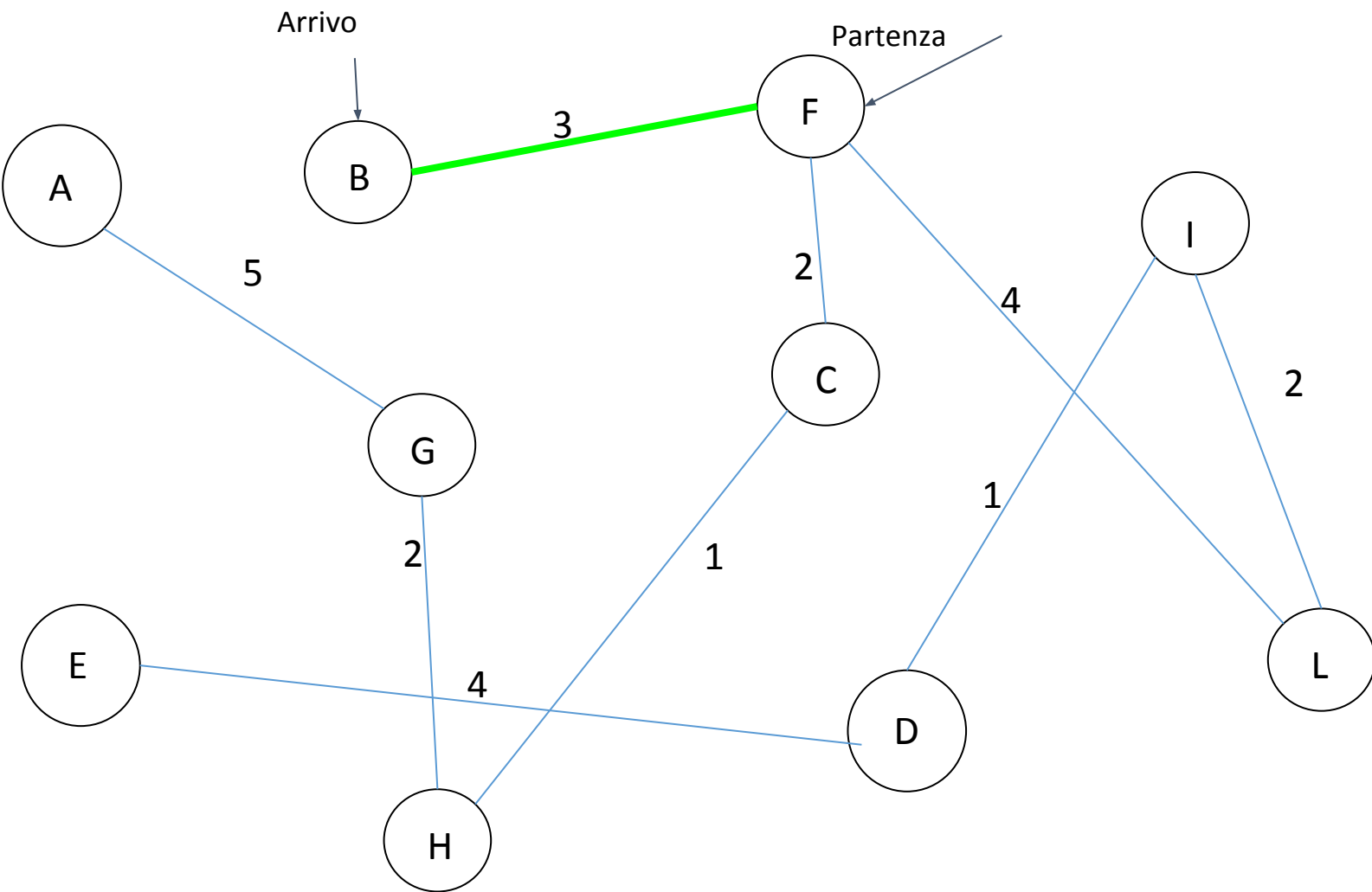




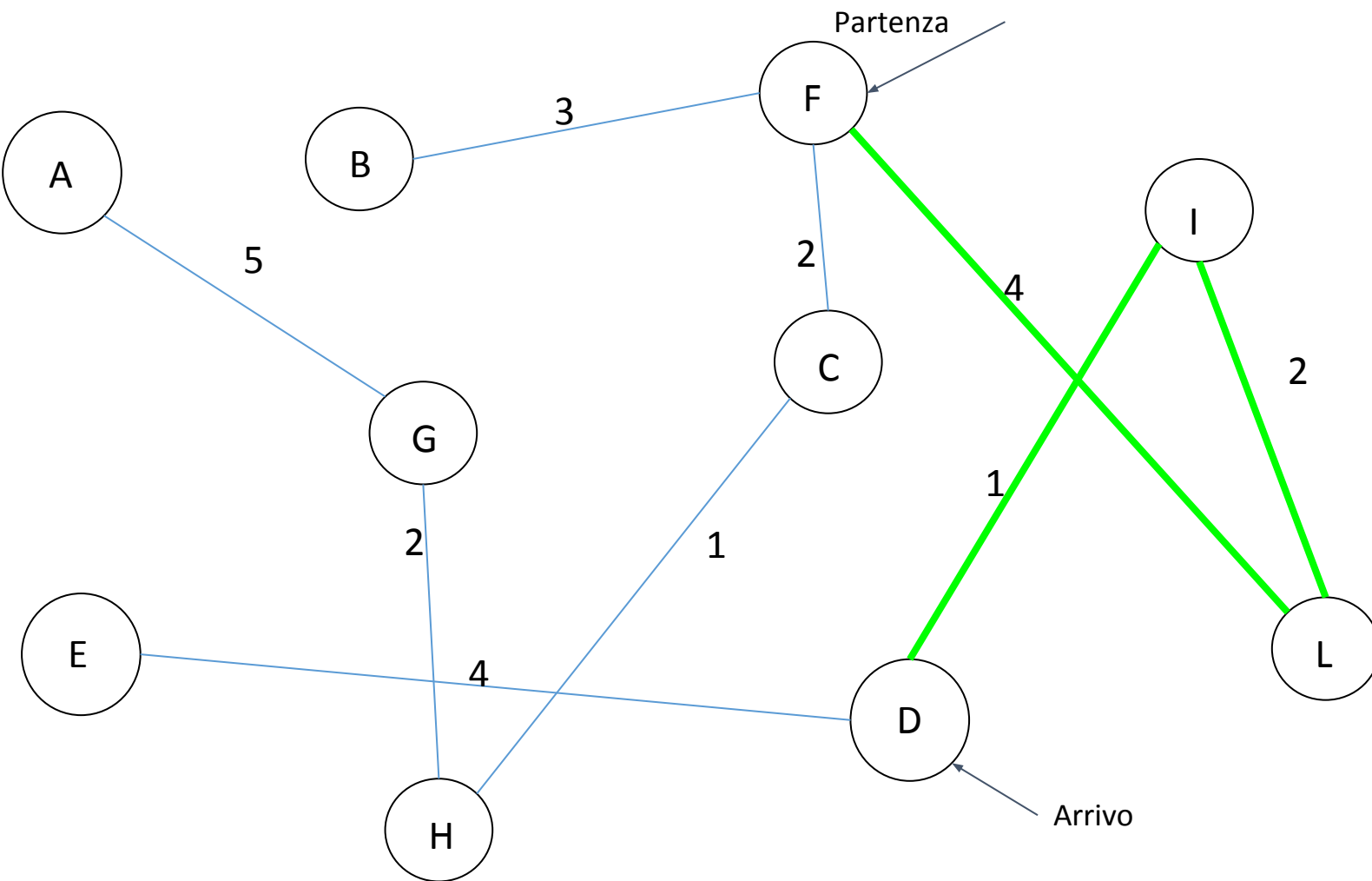
Vertici	Costo	DA
F	0	
C	2	F
B	3	F
H	3	C
L	4	F
G	5	H
I	6	L
D	7	I
A	10	G
E	11	D



Vertici	Costo	DA
F	0	
C	2	F
B	3	F
H	3	C
L	4	F
G	5	H
I	6	L
D	7	I
A	10	G
E	11	D



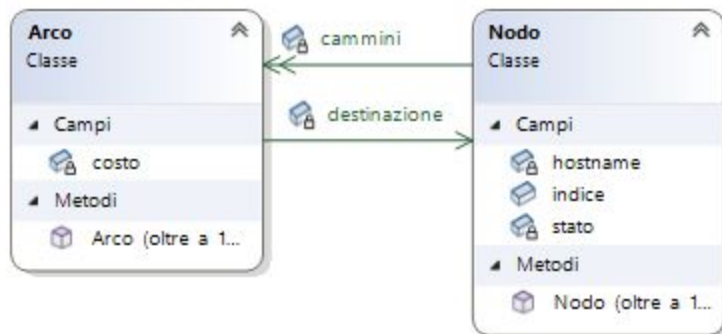
Vertici	Costo	DA
F	0	
C	2	F
B	3	F
H	3	C
L	4	F
G	5	H
I	6	L
D	7	I
A	10	G
E	11	D



Vertici	Costo	DA
F	0	
C	2	F
B	3	F
H	3	C
L	4	F
G	5	H
I	6	L
D	7	I
A	10	G
E	11	D

Sviluppo in C#

dell'algoritmo di Dijkstra



```
public class Nodo
{
    public static int indice; //numero di nodo del grafo
    bool stato;              // acceso spento
    string hostname;
    List<Arco> cammini;      // Lista degli archi uscenti dal nodo
}

public class Arco
{
    Nodo destinazione;
    int costo;
}
```

OSPF

Open Shortest Path First

Apri prima il percorso più breve

Il protocollo OSPF

Protocollo di routing robusto e scalabile. Offre una serie di funzionalità avanzate per la gestione efficiente delle reti IP

- **Link-State Protocol**
 - OSPF è un protocollo di routing basato sullo stato dei collegamenti.
 - Ogni router nella rete mantiene un database di stato dei collegamenti (**LSDB**) che contiene informazioni dettagliate sulla topologia di rete locale.
- **Metrica di Costo**
 - OSPF assegna un costo a ciascun collegamento in base alla sua velocità.
 - I router calcolano quindi i percorsi più brevi verso le destinazioni utilizzando questi costi come metrica.
- **Aree di OSPF**
 - OSPF organizza la rete in "aree" logiche, ciascuna con un router detto "router di confine dell'area" (ABR) che collega le aree tra loro.
 - Questa struttura gerarchica aiuta a ridurre la complessità della rete e migliorare l'efficienza del routing.
- **Convergenza Rapida**
 - OSPF supporta la convergenza rapida della rete, in cui i router sono in grado di rilevare rapidamente i cambiamenti nella topologia della rete e di aggiornare le loro tabelle di routing di conseguenza.
- **Autenticazione**
 - OSPF offre meccanismi di autenticazione per garantire che i messaggi di routing siano affidabili e provenienti da router autorizzati.

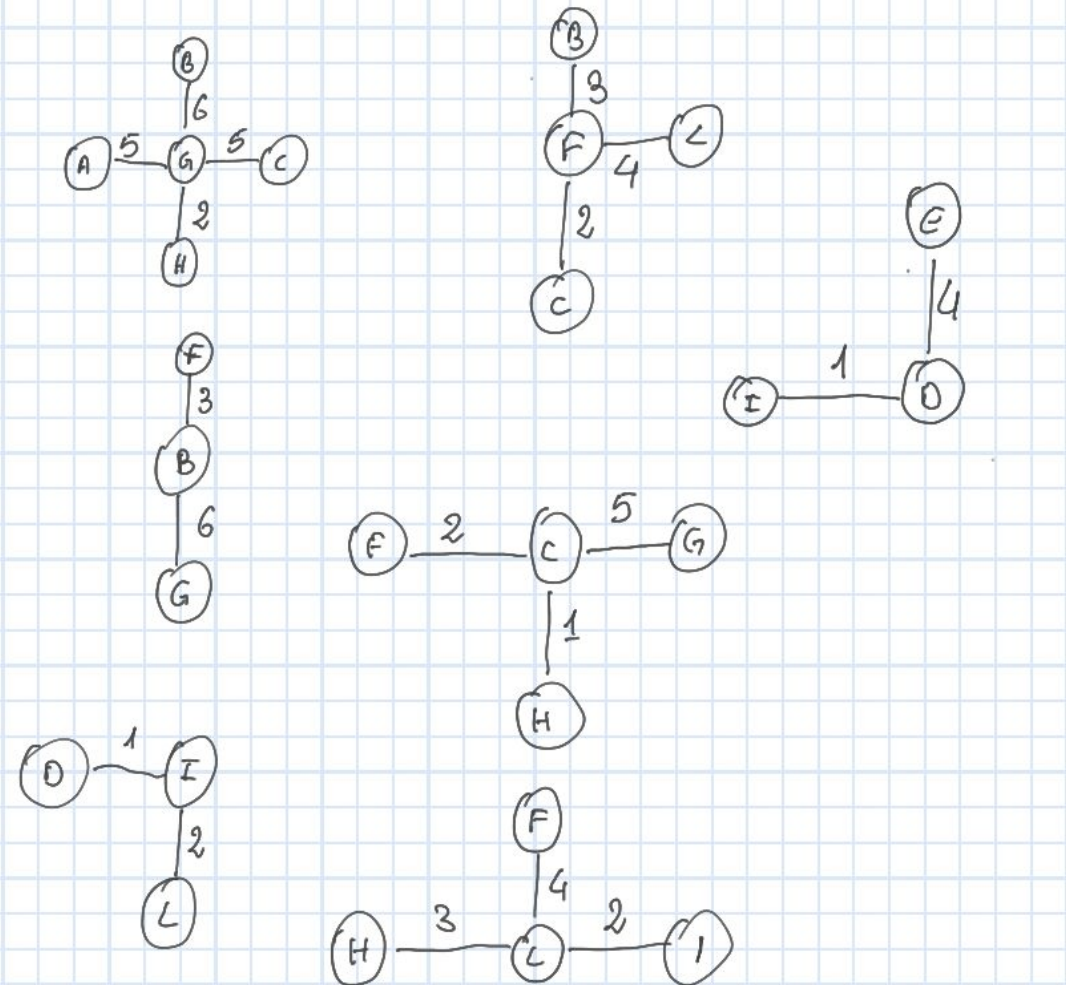
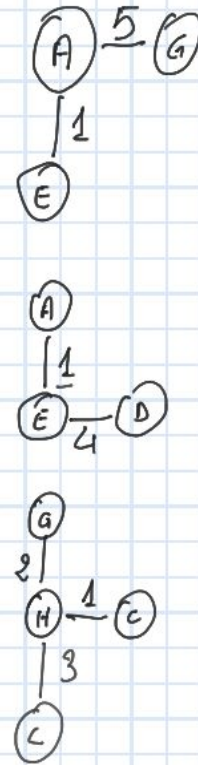
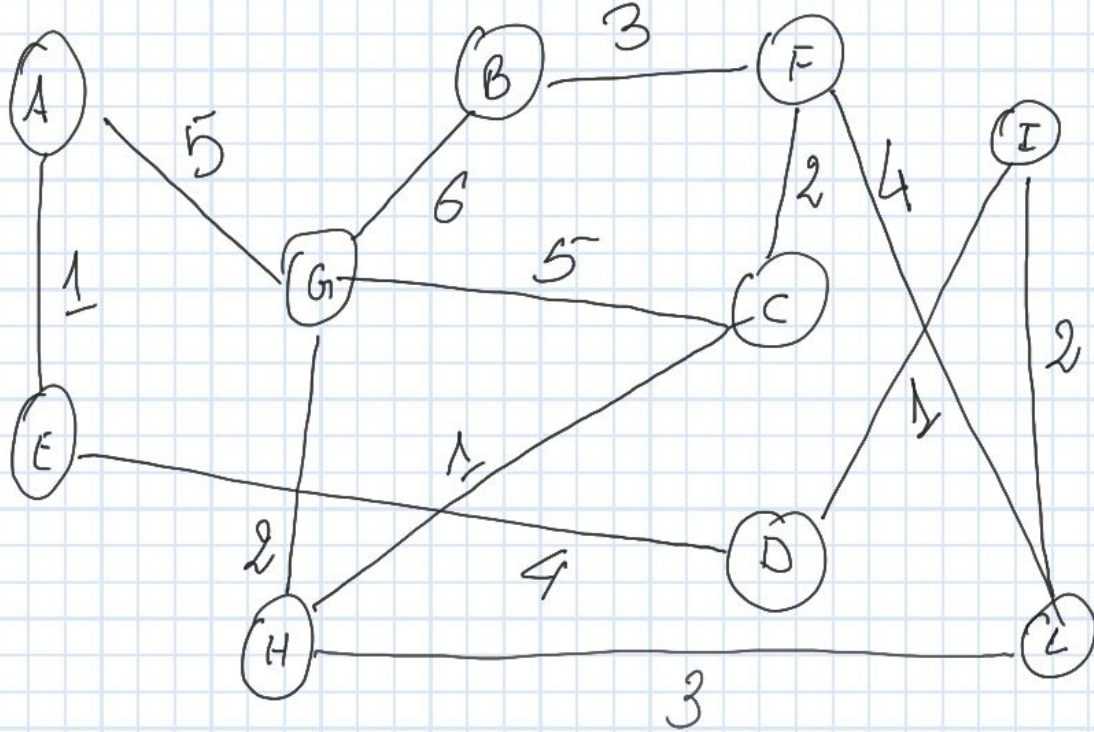
LSDB (Link-State Database)

"Database di Stato dei Collegamenti"

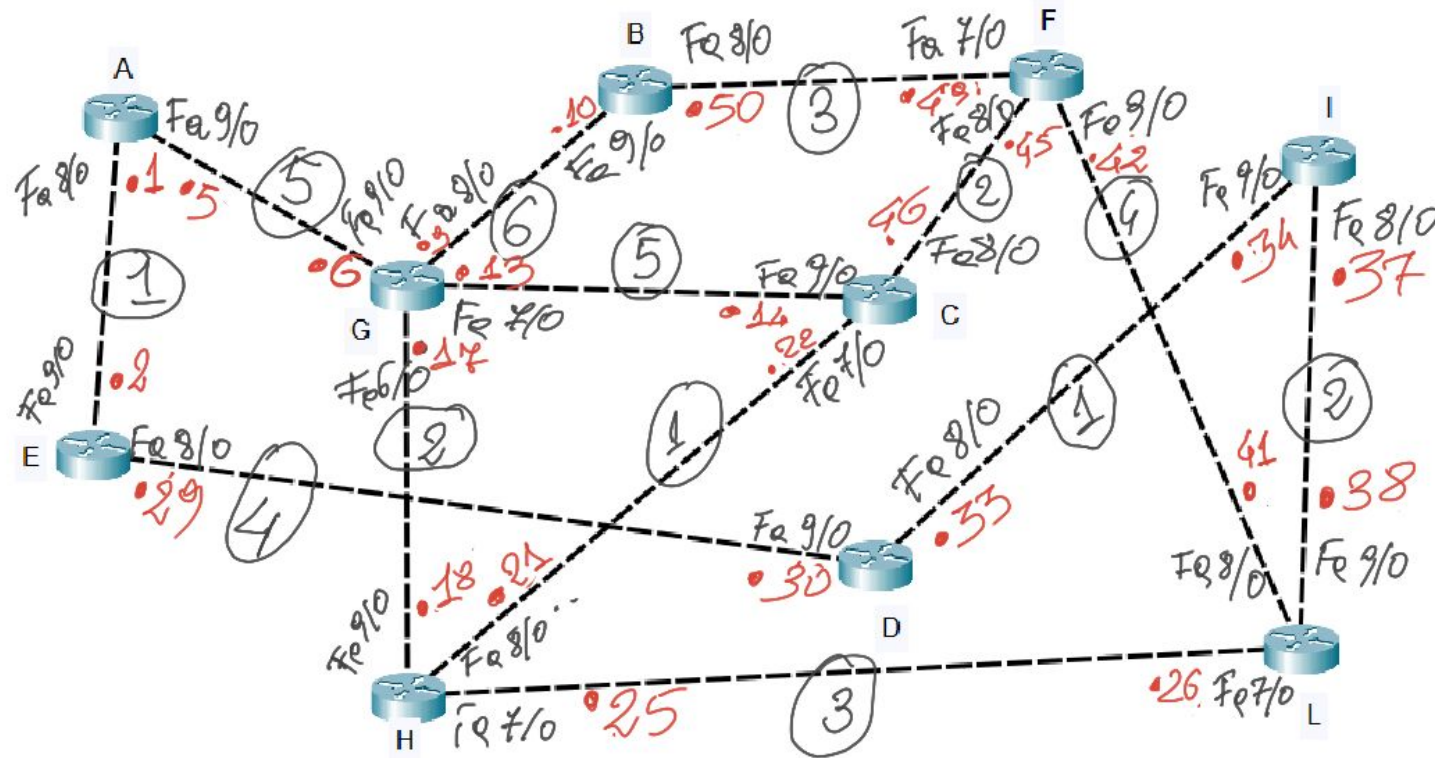
- LSDB
 - Contiene una rappresentazione della rete, indicando quali router sono presenti, quali reti sono collegate a ciascun router, e quali sono i costi associati ai collegamenti.
 - Queste informazioni vengono utilizzate per calcolare i percorsi più brevi verso tutte le destinazioni all'interno della rete.
- Avvio router OSPF
 - scambia messaggi di stato dei collegamenti con i router vicini
 - Aggiornamento delle informazioni sulla topologia di rete.

LSDB è una sorta di "mappa" della rete OSPF che ogni router utilizza per prendere decisioni di routing e per mantenere una conoscenza dettagliata della topologia di rete circostante

Ripartiamo dal Grafo



Area 0



Router A

```
Enable
configure terminal
hostname RT-A
```

```
router ospf 1
network 192.168.1.0 0.0.0.255 area 0
exit
```

```
interface fastethernet 8/0
ip address 192.168.1.1 255.255.255.252
no shutdown
ip ospf cost 1
exit
```

```
interface fastethernet 9/0
ip address 192.168.1.5 255.255.255.252
no shutdown
ip ospf cost 5
end
```