Lezioni di Ricerca Operativa

Corso di Laurea in Informatica

Università di Salerno

Lezione n° 23

- Algoritmo di Kruskal
- Algoritmo di Prim

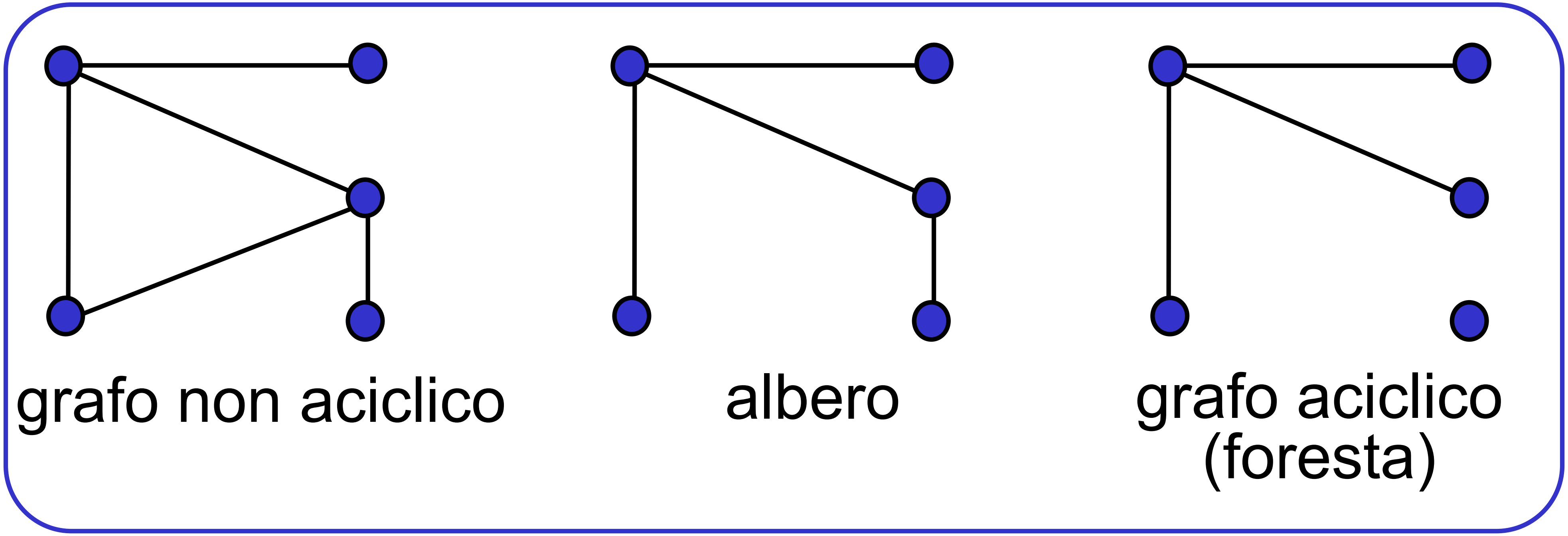
R. Cerulli – F. Carrabs

Alberi

Sia G=(V,E) un grafo non orientato e connesso.

- G è **aciclico** se i suoi archi non formano cicli;
- Un **albero** è un grafo connesso ed aciclico;
- Ogni grafo aciclico è in generale l'unione di uno o più alberi e viene detto **foresta;**

Esempio



Proprietà degli alberi

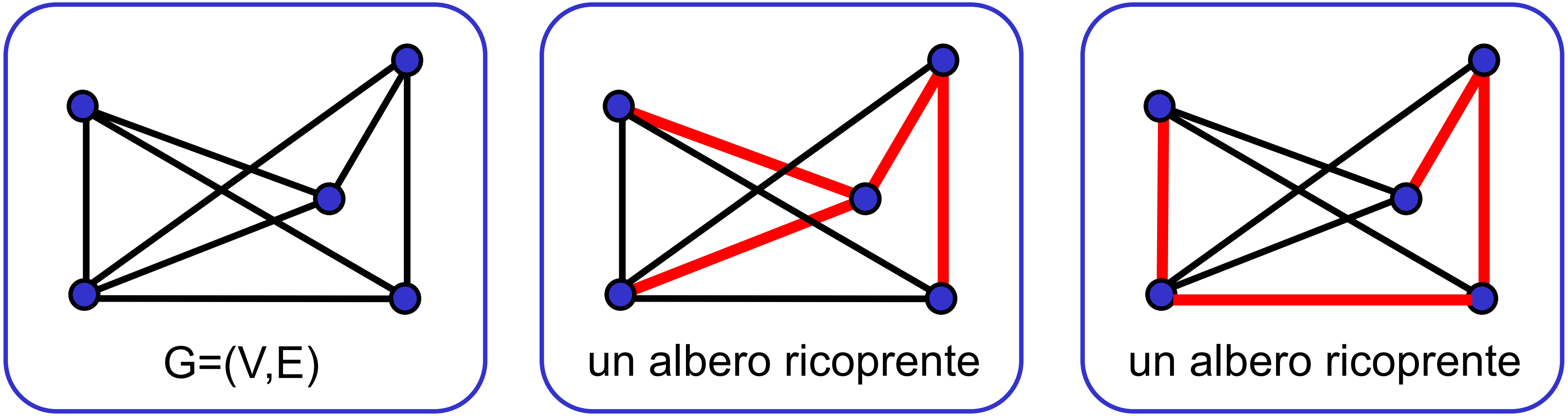
Dato un grafo G=(V,E), le seguenti affermazioni sono equivalenti:

- G è un albero
- ogni coppia di nodi di G è connessa da un unico cammino
- G è aciclico e $|E| = |V| - 1$
- G è connesso e $|E| = |V| - 1$

Alberi Ricoprenti

Sia G=(V,E) un grafo non orientato e connesso.

Un **albero ricoprente** (**spanning tree**) di G è un sottografo T di G tale che: T è un albero e contiene tutti i nodi di G.



Un grafo può avere più alberi ricoprenti.

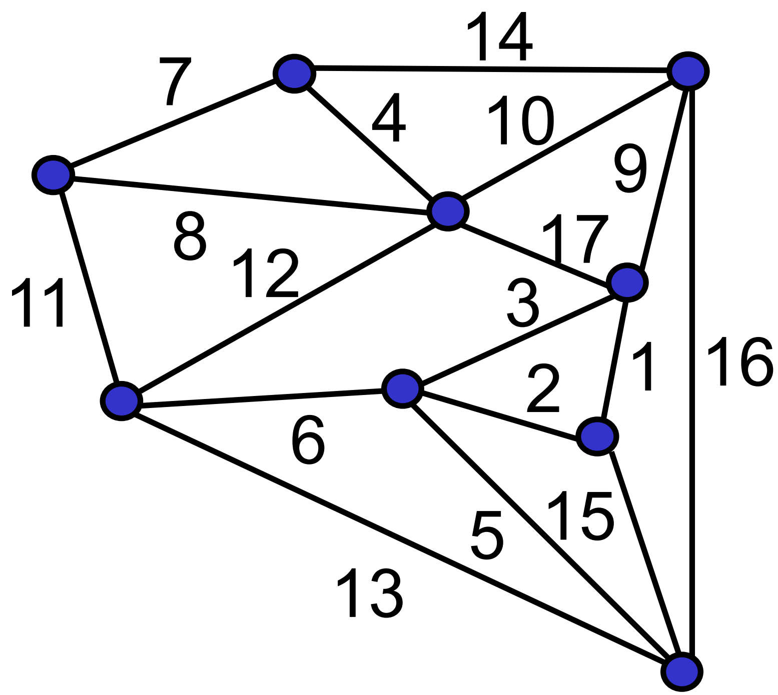
Il Problema del Minimo Albero Ricoprente (Minimum Spanning Tree Problem)

Sia G=(V,E) un grafo non orientato e connesso dove ad ogni arco $e_i \in E$ è associato un costo c_i.

Il **costo** di un albero ricoprente T di G è dato dalla somma dei costi degli archi che lo compongono.

Il problema: determinare l'albero ricoprente di G di costo minimo

Esempio



Applicazioni

- determinare la rete di comunicazione più affidabile;
- determinare la connessione tra n centri a costo minimo (e.g., distribuzione del gas);
- progettare i circuiti elettronici per collegare fra loro le diverse componenti minimizzando la quantità di filo da utilizzare;

Modello Matematico 1: Subtour Elimination

$$\min \sum_{(i,j)\in E} c_{ij} x_{ij}$$

$$\sum_{(i,j)\in E} x_{ij} = n - 1$$

$$\sum_{\substack{(i,j)\in E \\ i\in S, j\in S}} x_{ij} \leq |S| - 1 \qquad \forall \mathrm{S} \subset \mathrm{V},\ |S| \geq 3$$

$$x_{ij} \in \{0,1\} \qquad \forall\ (\mathrm{i,j})\ \in \mathrm{E}$$

Modello Matematico 1: Cut Formulation

$$\min \sum_{(i,j)\in E} c_{ij} x_{ij}$$

$$\sum_{(i,j)\in E} x_{ij} = \mathrm{n} - 1$$

$$\sum_{\substack{(i,j)\in E \\ i\in S,\, j\in V\setminus S}} x_{ij} \geq 1 \qquad \forall \mathrm{S} \subset \mathrm{V}, |S| \geq 1$$

$$x_{ij} \in \{0,1\} \qquad \forall\, (\mathrm{i}, \mathrm{j}) \in \mathrm{E}$$

Algoritmi Risolutivi

- algoritmo di Kruskal (Greedy Algorithm)
- algoritmo di Prim

Algoritmo di Kruskal (Minimum Spanning Tree)

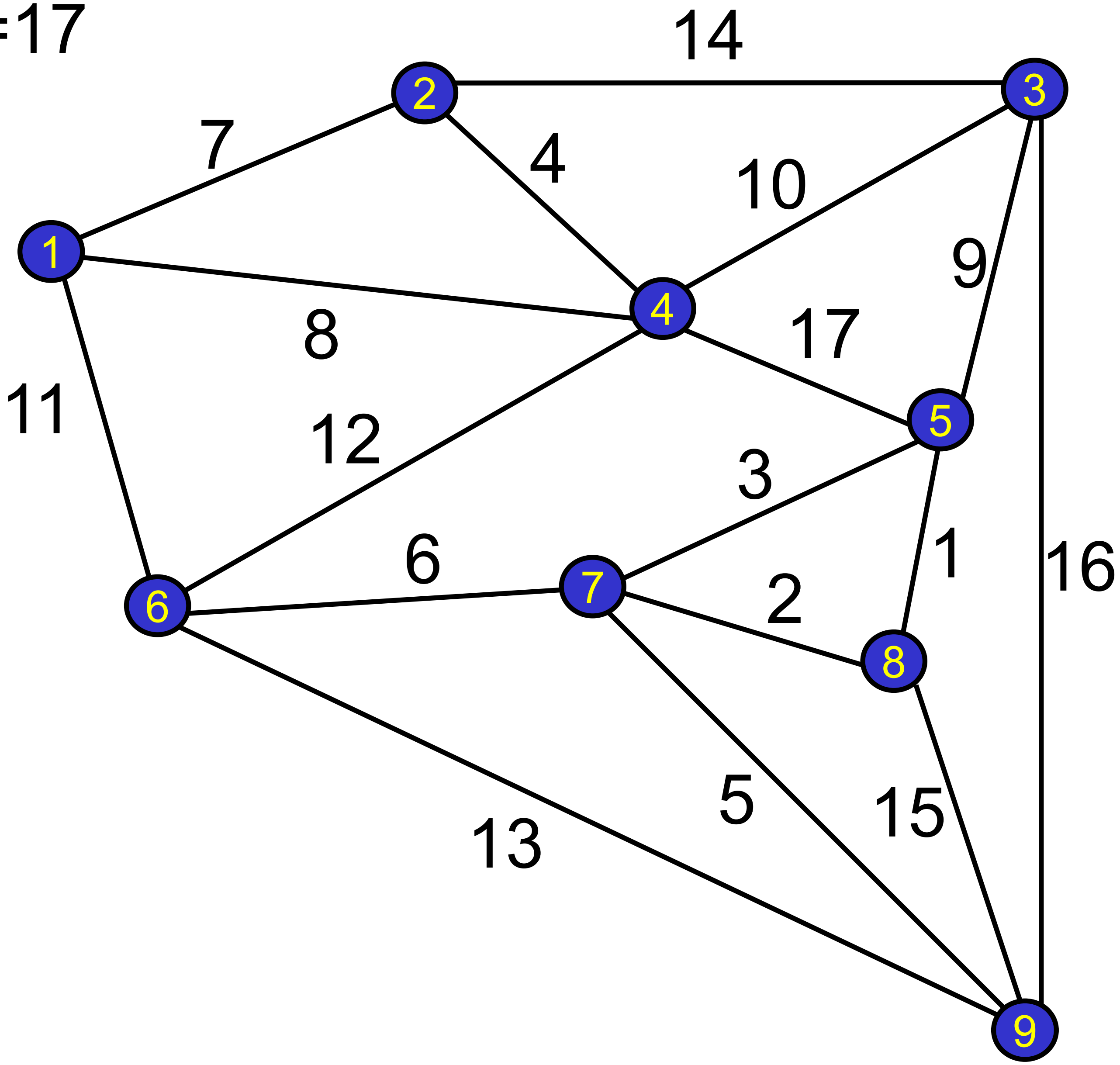
Sia G=(V,E) un grafo non orientato con *n* nodi ed *m* archi.

1. Ordinare gli archi $e_1, e_2, \ldots, e_m$ in modo non decrescente ($c_1 \leq c_2 \leq \ldots \leq c_m$). Siano $E^0=\emptyset$, $ST^0=(V, \emptyset)$ e k=1.

2. Se $(V, E^{k-1} \cup \{e_k\})$ è un grafo aciclico allora $ST^k=(V,E^k)$ con $E^k=E^{k-1} \cup \{e_k\}$, altrimenti e_k viene scartato, $E^k=E^{k-1}$ e $ST^k= ST^{k-1}$.

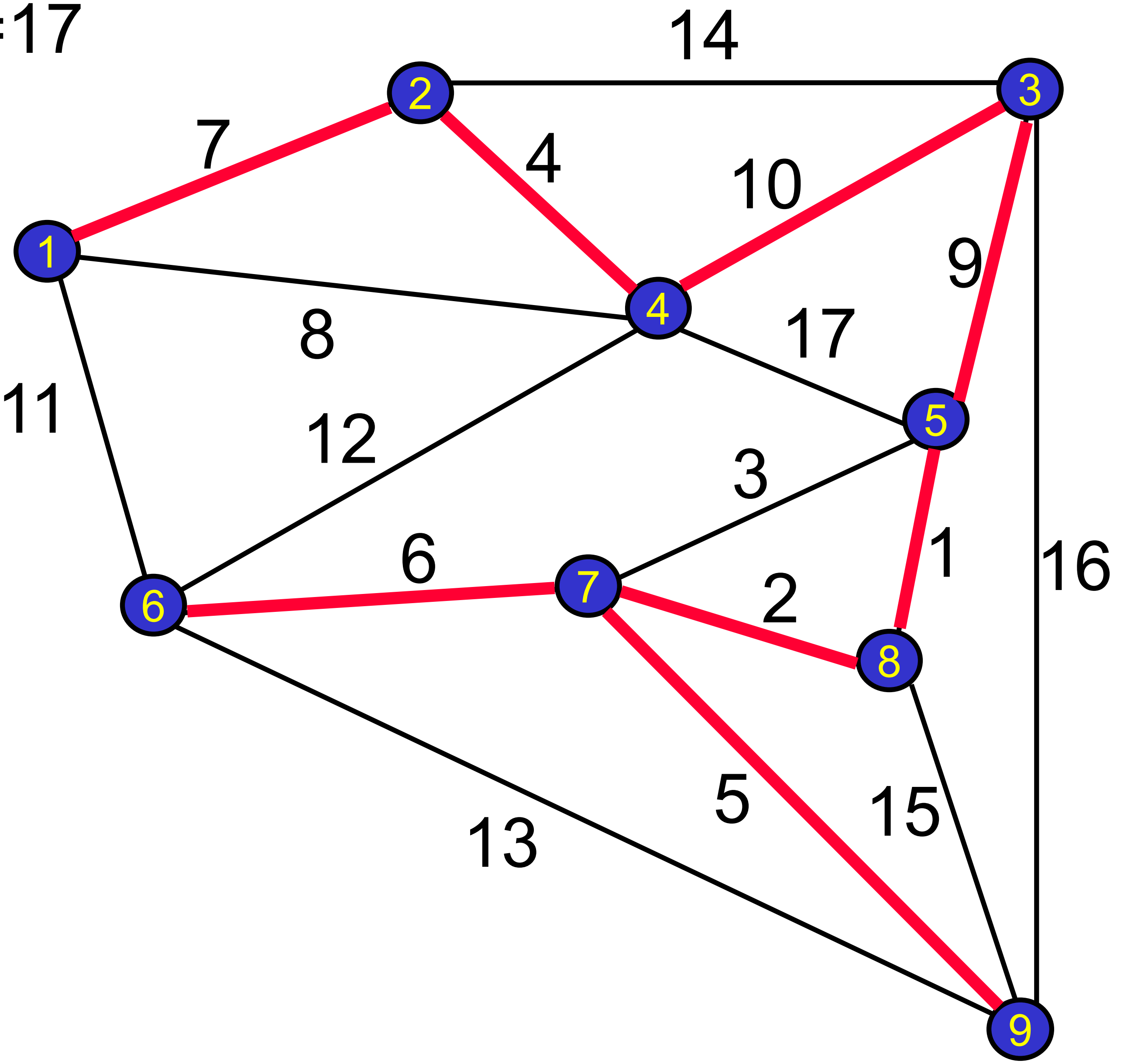
1. Se $|E^k|=n-1$ l'algoritmo si arresta ed ST^k è l'albero ricoprente cercato, altrimenti k=k+1 e si ritorna al passo (2).

Esempio: n=9 m=17



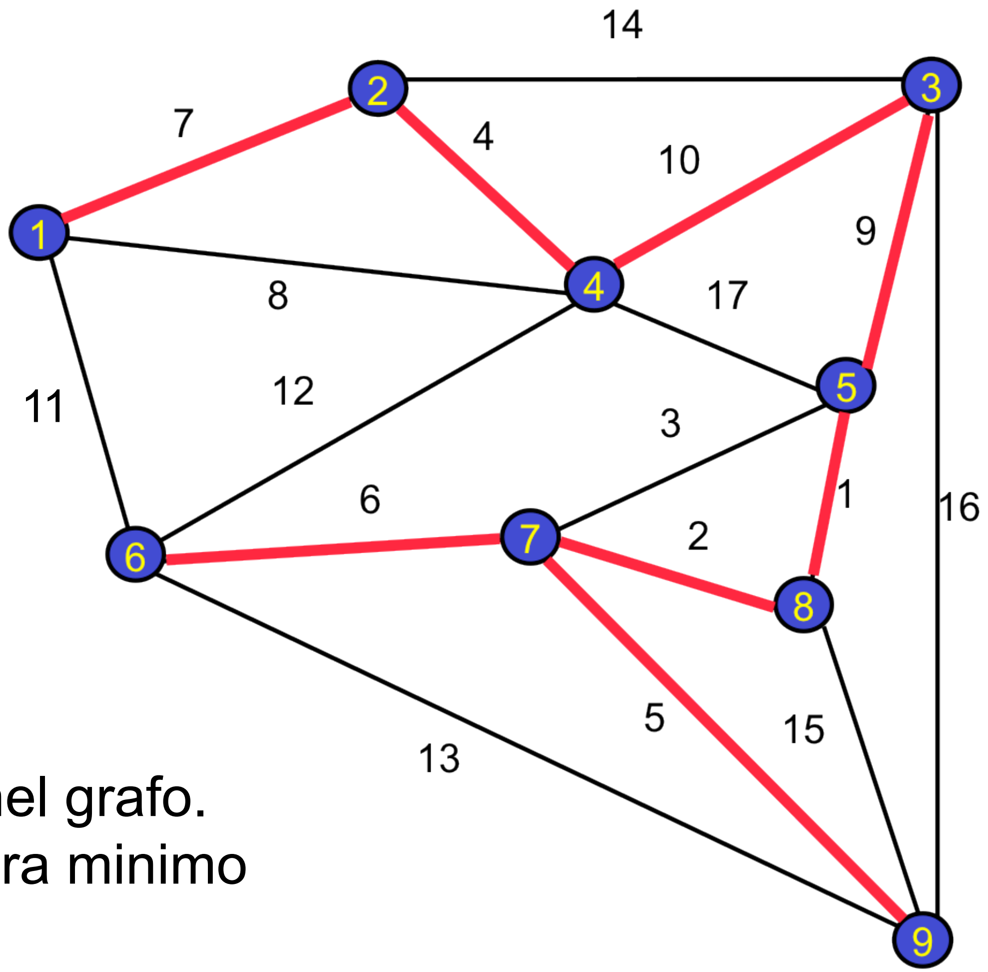
(5,8)	**(7,8)**	**(5,7)**	**(2,4)**	**(7,9)**	**(6,7)**	**(1,2)**	**(1,4)**	**(3,5)**	**(3,4)**	**(1,6)**	**(4,6)**	**(6,9)**	**(2,3)**	**(8,9)**	**(3,9)**	**(4,5)**
1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**

Esempio: n=9 m=17



~~(5,8)~~	~~(7,8)~~	~~(5,7)~~	~~(2,4)~~	~~(7,9)~~	~~(6,7)~~	~~(1,2)~~	~~(1,4)~~	~~(3,5)~~	~~(3,4)~~	(1,6)	(4,6)	(6,9)	(2,3)	(8,9)	(3,9)	(4,5)
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17

Esempio: n=9 m=17



- Sia 4+k il nuovo peso dell'arco (2,4) nel grafo. Per quali valori di k l'albero di copertura minimo non cambia?
- Sia 8+k il nuovo peso dell'arco (1,4) in G. Per quali valori di k l'arco (1,4) verrà inserito nella soluzione ottima?

~~(5,8)~~	~~(7,8)~~	~~(5,7)~~	~~(2,4)~~	~~(7,9)~~	~~(6,7)~~	~~(1,2)~~	~~(1,4)~~	~~(3,5)~~	~~(3,4)~~	(1,6)	(4,6)	(6,9)	(2,3)	(8,9)	(3,9)	(4,5)
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17

Algoritmo di Prim (Minimum Spanning Tree)

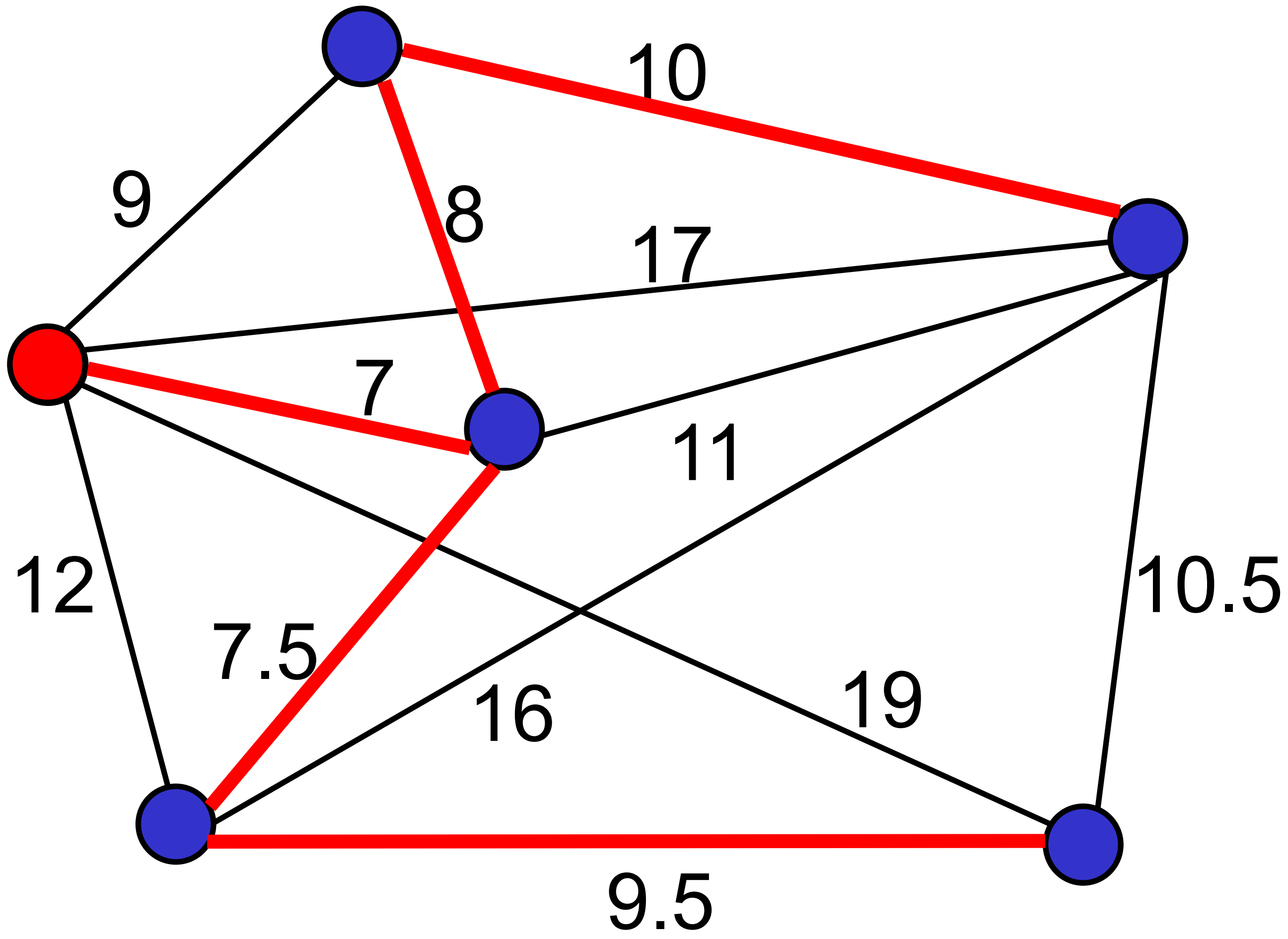
Sia $G=(V,E)$ un grafo non orientato con *n* nodi ed *m* archi.

1. Selezionare un qualsiasi vertice $v_s \in V$ e porre $V^0=\{v_s\}$, $E^0=\emptyset$, $k=1$.

2. Dato il taglio $[V^{k-1},V\backslash V^{k-1}]$, selezionare l'arco diretto del taglio (v_i,v_h) avente costo minimo e porre $V^k=V^{k-1}\cup\{v_h\}$ e $E^k=E^{k-1}\cup\{(v_i,v_h)\}$

1. Se $|E^k|=n-1$ l'algoritmo si arresta ed $ST=(V^k,E^k)$ è l'albero ricoprente cercato, altrimenti $k=k+1$ e si ritorna al passo (2).

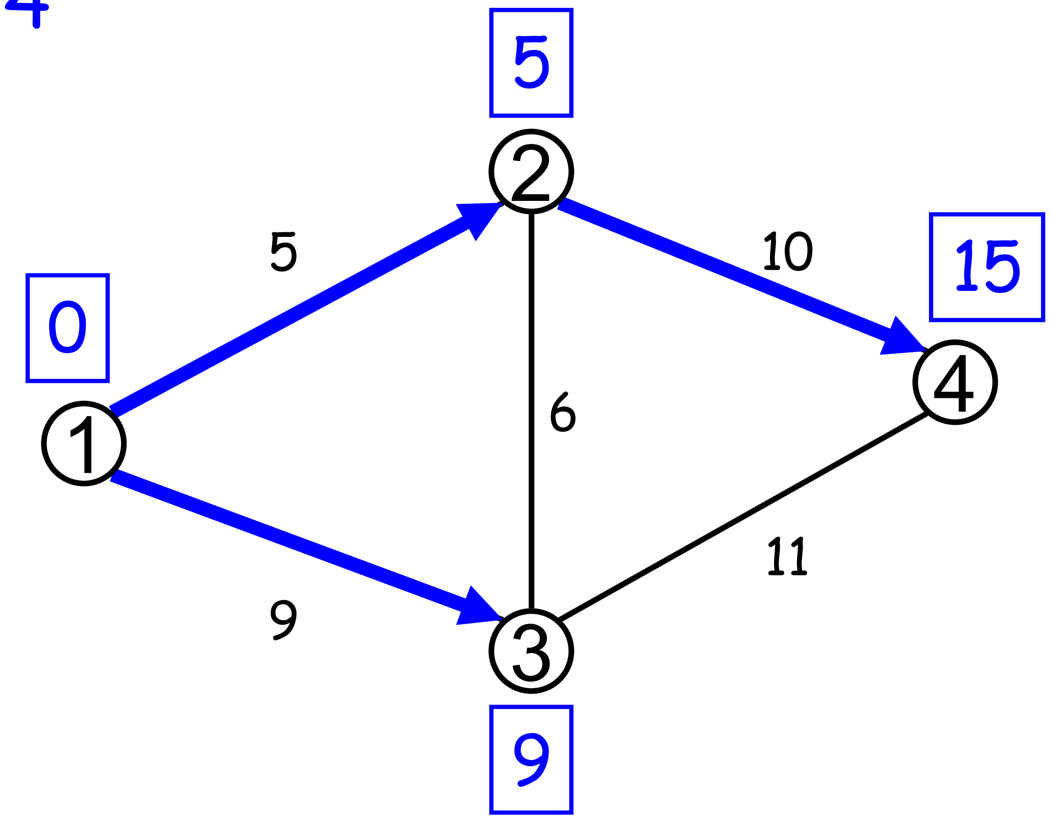
Esempio: n=6 m=12



L' algoritmo di Prim O(|E|log|V|) è più efficiente di quello di Kruskal O(|E|log|E|).

Albero dei Cammini minimi ≠ albero di copertura minimo

SPT=24



Albero dei Cammini minimi ≠ albero di copertura minimo