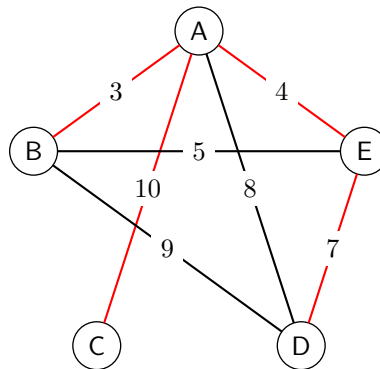


ale-cci

Modelli Algoritmi per il Supporto alle Decisioni

March 31, 2020

Minimum Spanning Tree



Algoritmo Greedy

```
import utils

graph = [(10, 'A', 'C'), (8, 'A', 'D'), (7, 'D', 'E'),
         (4, 'A', 'E'), (3, 'B', 'A'), (9, 'B', 'D'), (5, 'B', 'E')]
N = utils.vertices_of(graph)

connected = set()
mst = []

edges = sorted(graph)
for edge in edges:
    weight, lhs, rhs = edge

    # Two nodes already connected
    if lhs in connected and rhs in connected:
        continue

    mst.append(edge)
    connected.update({lhs, rhs})

    if len(mst) == N:
        break
```

Correttezza algoritmo Greedy

Supponiamo per assurdo che esista un' diverso MST $T' = (V, E_{T'})$ di peso inferiore a $T = (V, E_T)$, quello restituito dall'algoritmo greedy.

Siccome i due alberi hanno costo diverso, differiscono di almeno un' arco. Indichiamo con e_h l'arco a peso minore appartenente a $\{E_T - E_{T'}\}$. Dato che T' è un MST, esiste un ciclo C in $\{e_h\} \cup E_{T'}$ contenente l'arco e_h . Siccome anche T è un albero, quindi non ha cicli, allora $C \cap E_T \neq \emptyset$. Chiamiamo e_r l'arco a peso minore appartenente a $C \cap \{E_T - E_{T'}\}$. Necessariamente $w_{e_r} \leq w_{e_h}$, altrimenti l'algoritmo greedy applicato a T avrebbe selezionato prima e_h al posto di e_r . Sostituendo in T' l'arco e_r con e_h ottengo un nuovo albero di peso inferiore.

Questo va contro l'ipotesi T' è l'albero di supporto a peso minore.

Analisi complessità algoritmo greedy

$O(E \cdot \log(E))$, dovuta all'ordinamento degli archi in ordine di peso. Il controllo dell'esistenza di cicli è effettuato in $O(1)$.

Foresta di supporto

Viene chiamata foresta di supporto di un grafo G un grafo parziale $F = (V, E_F)$ privo di cicli. In particolare, un albero di supporto è una foresta con una sola componente connessa.

Contents

Minimum Spanning Tree

1