

Algoritmo di ricerca di flusso massimo tramite Edmond-Karp senza alcuna ottimizzazione

Filippo Magi

December 23, 2021

1 Algoritmo senza ottimizzazione

Algorithm 1 Ricerca del massimo flusso a costo minimo senza alcuna ottimizzazione

Require: grafo dei residui $\vec{G} = \{V(G), E(G) \cup \{\vec{e} : e \in E(G)\}\}$.

Ensure: valore del flusso massimo

$s \leftarrow \text{SourceNode di } \vec{G}$

$t \leftarrow \text{SinkNode di } \vec{G}$

loop

$f \leftarrow \text{DoBfs}(\text{grafo})$

if $f = 0$ **then**

break

end if

$\text{mom} \leftarrow t$

while $\text{mom} \neq s$ **do**

 aggiorno capacità o flusso del nodo m a seconda del suo predecessore

$\text{mom} \leftarrow \text{predecessore di mom}$

end while

end loop

return flusso uscente da s

Algorithm 2 Algoritmo DoBfs senza alcuna ottimizzazione

Require: grafo dei residui $\vec{\bar{G}}$

Ensure: quantità di flusso inviata, grafo dei residui $\vec{\bar{G}}$ aggiornato per cammino f aumentante

cancella informazioni precedenti presenti in $V(G)$

coda \leftarrow Coda di nodi

coda.Enqueue(SourceNode di $\vec{\bar{G}}$)

while coda non è vuota **do**

 element \leftarrow coda.Dequeue

for all arco **edge** che esce dal nodo element **do**

 n \leftarrow nodo che entra da edge

if nodo n non è già stato visitato AND è possibile inviare(o ritirare) del flusso **then**

 aggiorno dati di n (nodo precedente, label, flusso entrante)

if n è il SinkNode di $\vec{\bar{G}}$ **then**

return flusso entrante in n

else

 coda.Enqueue(n)

end if

end if

end for

end while

return 0
