

RIPASSO DIJKSTRA

DJKSTRA(s)

Initialize-single-source(s)

$Q := \text{Build-heap}(V, d)$

While $Q \neq \emptyset$ do

$v := \text{Extract-min}(Q)$

SCAN(v)

Initialize-single-source(s)

for $v \in V$

$d[v] = +\infty$

$\text{Pred}[v] = \text{NIL}$

$d[s] = 0$

SCAN(v)

for $u \in \text{Adj}[v]$

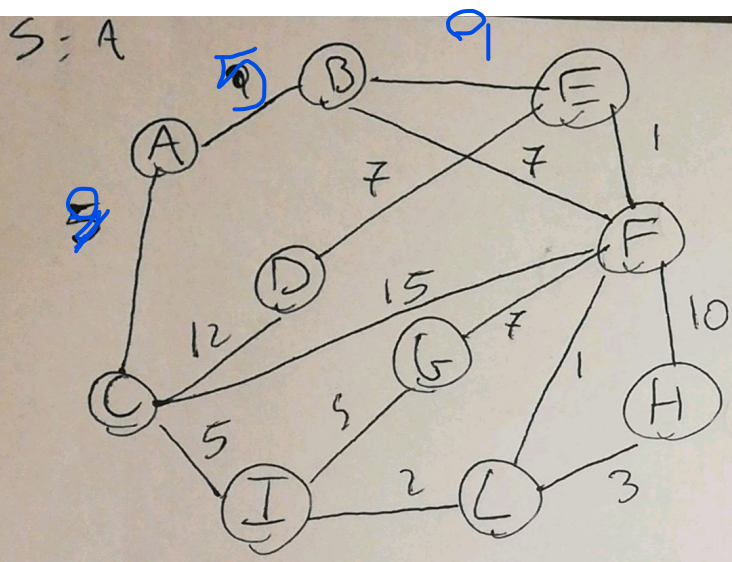
Relax(v, u)

Relax(v, u)

if $d[u] > d[v] + w(v, u)$

$Q.\text{decreaseKey}(d[u], d[v] + w(u, v))$

$\text{Pred}[u] = v$



Si crea vettore dei Pred lungo $|V|$ e vettore di dist A
Pred

X	X	X	X	X	X	X	X	X	X	X
A	B	C	D	E	F	G	H	I	L	

A

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

La initialize mettere A tutto a infinito tranne A a 0 e Pred tutto a NIL x
Si organizza una coda di priorità dei nodi che li gestisce con Key = d
Nel ciclo selezioniamo (estrainiamo) il minimo e applichiamo su tutti i suoi adiacenti la formula

Estraiamo A (non c'è già nella coda)
 $(B) \quad (A) \quad (w(A,B))$
 $9 > 0 + 9$ TRUE: $d[B] \leftarrow 0 + 9$
Pred[B] \leftarrow A
 FALSE: nulla

Facciamo così anche per C
e poi inizia la prossima iterazione col nuovo min

Output per questo grafo e S = A

Pred

X	A	A	E	F	B	I	L	C	F
A	B	C	D	E	F	G	H	I	L

d

0	5	9	20	13	12	18	16	15	13
---	---	---	----	----	----	----	----	----	----

G: ACIG
D: ABFED

B: AB
C: AC
F: ABF
L: ABFL
E: ABFE
I: ACI
H: ABFLH