Esercizio 20

~~AL~~ ALGO $(G=(V,E), s)$

~~Poni~~ $S=\{s\}$ $d[s]=0$ $d'[v]=\infty$ $\forall v \in V-\{s\}$

while $S \neq V$

prendi elemento $v \in V-S$ che ha un elemento in S che sia

$d'[v] = \min_{e=(u,v):\, u \in S} d[u] + l(e)$ del più piccolo possibile

poni $d[v] = d'[v]$ aggiungi v a S

Per implementare quest'algoritmo utilizziamo la tecnica Greedy, ed ad ogni passo scegliamo sempre il cammino minimo più vicino, siamo sicuri che l'algoritmo sceglie sempre il cammino vicino ~~...~~.
Preso un cammino ~~...~~ minimo, siamo sicuri che è il minimo, perché se non fosse così, allora non sarebbe stato proprio scelto.

Usando code a priorità con n vertici ed m archi, l'algoritmo richiede tempo $O(m)$, che aggiunge il tempo per n DecreaseKey e n ExtractMin. Se per le code a priorità usiamo un min-heap, allora le op. richiedono tempo $O(\log n)$, il tempo complessivo sarà $O(m \log n)$

Esercizio 33

PRIM $(G=(V,E), s)$

$Q=\{s\}$ $a(s)=0$ $a(x)=\infty$ $\forall x \in V$ parent$(s) \leftarrow$ NIL

AFFINCHE $Q \neq \emptyset$

Estrai v da Q

per ogni arco $e=(u,v) \in E$ con $u \in Q$

if $c(e) < a(u)$

$a(u) \leftarrow c(e)$

parent$(u) = v$

decreasekey$(Q, u, a(u))$