

Algoritmo di Kruskal

Trova l'albero di copertura di costo minimo.

1. Trova la sequenza ordinata in ordine di costo crescente degli archi del grafo (a_1, \dots, a_m) . $T = \emptyset$, $k = 1$;
2. se $|T| = n - 1$, termina l'esecuzione;
3. se a_k non ha forma un ciclo con gli archi di T , $T = T \cup \{a_k\}$;
4. $k = k + 1$, torna al passo 2;

La soluzione è ottima per la condizione di ottimalità basata sui cicli.

Per determinare se aggiungere un arco forma un ciclo, assegnamo ad ogni nodo un'etichetta d_i inizializzata a i , che rappresenta la componente connessa di cui fa parte. (u, v) crea un ciclo se $d_u = d_v$. Quando aggiungiamo (u, v) a T , assegnamo $\min\{d_u, d_v\}$ a tutti i nodi i con $d_i = \max\{d_u, d_v\}$.

Complessità

$O(E \log E + EV) = O(E(\log E + V))$ (ordinamento più aggiornamento delle etichette), ma si può raggiungere $O(E \log V)$ con un'implementazione efficiente di un disjoint set per determinare se un arco forma un ciclo.