

Algoritmo ricorsivo che cerca il flusso massimo tramite una ricerca in profondità.

1 Strutture dati utilizzate

1.1 BiEdge

I nodi vengono collegati tra di loro da archi BiEdge, che contiene le informazioni da quale nodo esce e in quale nodo entra. Ovviamente conserva in memoria la quantità di flusso che passa e la sua capacità residua. Durante l'inizializzazione, qualsiasi nodo tranne il nodo destinazione t , avrà distanza pari a -1, per indicare che non è stato ancora esplorato.

1.2 Node

Ovviamente ogni nodo tiene una lista con tutti gli archi a lui collegati, si salva il nodo e l'arco predecessore per indicazioni su dove inviare il flusso. Inoltre, si salva la distanza tra quel nodo e t .

1.3 Graph

Un insieme di nodi.

2 Descrizione algoritmo

Eseguo una bfs, esplorando tutti i nodi. Invio il flusso che ho trovato grazie a lei. Cancello le informazioni riguardanti le indicazioni su tutti i nodi (elimino le informazioni su previousNode e previousEdge). dopo aver ottenuto le distanze, finché la distanza tra s è minore del numero di nodi del grafo, proseguo con l'algoritmo ricorsivo di ShortestAugmentingPath (nel codice e pseudo-codice, Dfs), e con il percorso trovato, invio il flusso massimo consentito.

2.1 bfs

iniziamo con una bfs da t esplora tutti i nodi, indicando la distanza da lui, oltre a trovare un percorso da t e il nodo sorgente s .

2.2 dfs

Algoritmo ricorsivo che cerca di procedere da s verso t , attraverso procede, cioè se il nodo \mathbf{p} che sto analizzando ha un arco che lo collega con un nodo \mathbf{n} tale che $\mathbf{p}.distance = \mathbf{n}.distance + 1$, e con capacità positiva, allora procedo ad salvare in una apposita coda e analizzare \mathbf{n} , a meno che non sia il nodo cercato t (che verrà comunque salvato nella coda). Se non è possibile procedere, procedo a fare il retreat, cioè cerco tra i nodi che escono da \mathbf{p} quello con distanza minore,

e rendo la distanza di \mathbf{p} il valore di quella distanza $+1$, per poi analizzare il nodo che avevo analizzato antecedente, nel caso il non lo abbia (cioè \mathbf{p} è il nodo sorgente s), ripeto con \mathbf{p} . Appena trovo il nodo t , invio il flusso trovato. Per ogni nodo esplorato, lo inserisco in una coda, in maniera tale, dopo aver inviato il flusso, di cancellare tutti i dati dei nodi esplorati.