

1. Divide et Impera

- a. [9 cfu vecchio] Si scriva lo pseudocodice dell'algoritmo QuickSelect. Descrivere il comportamento di eventuali algoritmi invocati da Quickselect.

2. Grafi

- d. [6 cfu nuovo] Si scriva l'algoritmo ricorsivo che effettua la visita DFS di un grafo. Analizzare il tempo di esecuzione dell'algoritmo.

3. Algoritmi greedy

- d. [6 cfu nuovo] Su quali fatti si basa la dimostrazione dell'ottimalità della strategia Farthest-In-Future? Spiegare perché i suddetti fatti implicano che la strategia Farthest-In-Future produce la soluzione ottimale.

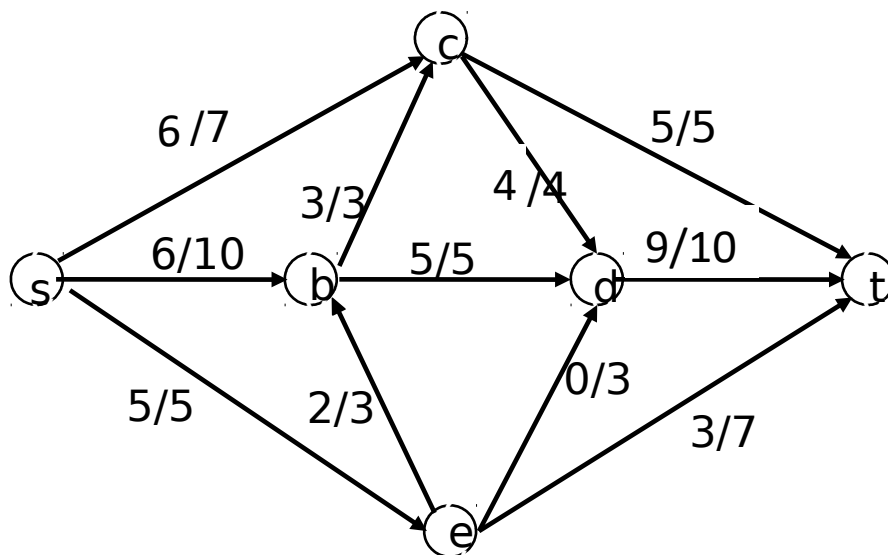
4. Massimo flusso

[6 cfu vecchio e 9 cfu vecchio]

a) Si consideri la seguente rete di flusso e la funzione di flusso i cui valori sono indicati a sinistra delle capacità degli archi. Si disegni la rete residua rispetto alla funzione di flusso indicata e si dica se questa funzione ha valore massimo. Nel caso in cui la funzione non abbia valore massimo, si fornisca **la funzione flusso con valore massimo e il taglio di capacità minima**. A tal fine si eseguano una o più iterazioni dell'algoritmo di Ford-Fulkerson a partire dalla funzione di flusso data.

Per ogni iterazione dell'algoritmo, occorre

- disegnare la rete residua all'inizio di quell'iterazione
- indicare il cammino aumentante da voi scelto
- mostrare il valore associato ad ogni arco del grafo al termine di quella iterazione



- b) [6 cfu vecchio] Si dimostri che il massimo matching di un grafo bipartito ha dimensione maggiore o uguale del valore del massimo flusso per la rete di flusso costruita a partire dal grafo bipartito. Spiegare anche come si costruisce la suddetta rete di flusso.
- c) [6 cfu vecchio e 9 cfu vecchio] Si scriva lo pseudocodice dell'algoritmo Augment invocato dall'algoritmo di Ford-Fulkerson. Analizzare il tempo di esecuzione nel caso pessimo dell'algoritmo Augment.

Progettazione di Algoritmi (domande aggiuntive 9 cfu vecchio e 6 cfu)
9/7/2020