Progettazione di Algoritmi
Anno Accademico 2016/2017
Appello del 9/6/2017

Cognome e Nome:
Numero di Matricola:

Spazio riservato alla correzione

1	2	3	4	Bonus	Totale
/25	/25	/25	/25		/100

1. Grafi

a) Si scriva lo pseudocodice dell'algoritmo ricorsivo DFS con l'aggiunta della linea di codice che serve per costruire l'albero DFS T . L'albero DFS T e` mantenuto in un'area di memoria accessibile globalmente ed e` inizializzato all'esterno della DFS con l'insieme vuoto. Si analizzi il tempo di esecuzione dell'algoritmo proposto. Analizzare il tempo di esecuzione significa fornire un limite superiore asintotico quanto migliore e` possibile al tempo di esecuzione dell'algoritmo giustificando la risposta.

b) Descrivere un algoritmo che, dato in input un grafo **non direzionato** G, trova tutte le componenti connesse di G. Si dica quali accorgimenti bisogna utilizzare affinche' l'algoritmo non visiti piu` volte gli stessi nodi e quale struttura dati utilizzare per riuscire a ricostruire le componenti connesse al termine dell'algoritmo.

c) Si analizzi il tempo di esecuzione dell'algoritmo proposto al punto b). Analizzare il tempo di esecuzione significa fornire un limite superiore asintotico quanto migliore e` possibile al tempo di esecuzione dell'algoritmo giustificando la risposta.

2. Algoritmi greedy
    a. Si descriva in modo chiaro e schematico in che cosa consiste un'istanza del problema del partizionamento di intervalli e qual e` l'obiettivo del problema. Se dalla risposta a questo punto si evincera` che lo studente non sa in cosa consiste il problema del partizionamento di intervalli, i punti successivi dell'esercizio non saranno valutati.

b. Si fornisca un esempio di istanza del problema del partizionamento di intervalli con n=6 intervalli e profondita` d= 2 e si distribuisca l'input sul piano in modo da evidenziare come e` fatta la soluzione ottima.

c. Si fornisca un algoritmo ottimo greedy per il problema del partizionamento di intervalli.

d. Qual e` il tempo di esecuzione dell'algoritmo? Giustificare la risposta spiegando quali accorgimenti occorre adottare nell'implementazione per ottenere quel tempo di esecuzione?

[**Bonus** (solo se si e` risposto correttamente al punto c dell'esercizio 2)] Si dimostri che l'algoritmo greedy di cui al punto b. usa esattamente un numero di risorse pari a ...

3. Programmazione dinamica.
   Si consideri il seguente problema.
   Il vostro medico vi ha detto che il vostro fabbisogno energetico giornaliero e` di 2000 Kcal al giorno e che non dovete superare questo numero di calorie. Il medico vi ha quindi fornito un elenco di 100 alimenti di ognuno dei quali potete mangiare **un'unica porzione al giorno** e per ciascuno dei quali e` indicato **l'apporto calorico per porzione**. A ciascun elemento dell'elenco avete assegnato un voto che esprime il vostro gradimento per l'alimento.

   Volete scegliere dall'elenco gli alimenti da mangiare in una giornata tipo in modo da rispettare la prescrizione del medico e in modo da soddisfare quanto piu` e` possibile i vostri gusti (in modo cioe` da massimizzare la somma dei voti degli alimenti selezionati). Ricordate che di ogni alimento potete mangiare un'unica porzione al giorno.

   a. Fornire una formulazione del problema sotto forma di problema computazionale specificando in modo chiaro e formale in cosa consiste un'istanza generica del problema e qual e` l'obiettivo del problema.

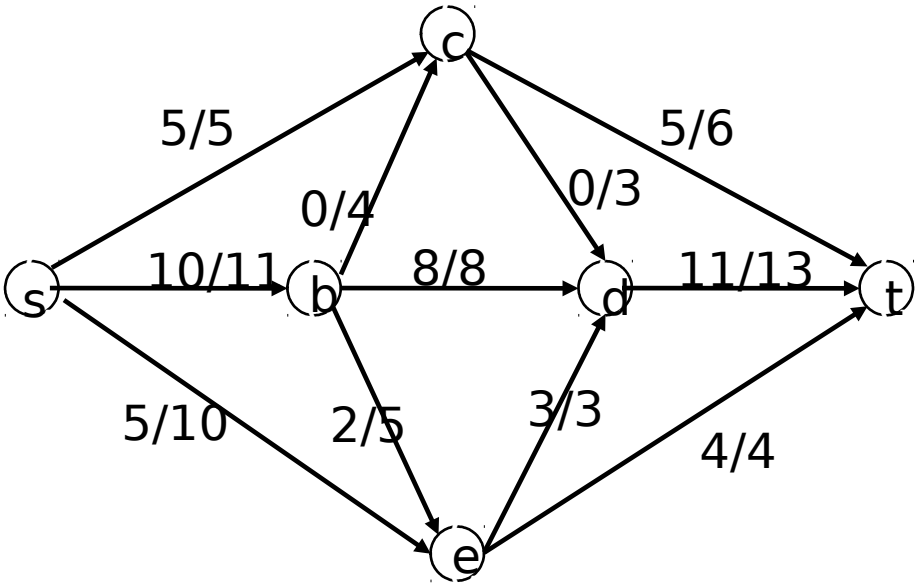
b. Fornire una formula per il calcolo del valore della soluzione ottima per il problema illustrato. Spiegare in modo chiaro e schematico come si arriva alla formula da voi fornita definendo in modo chiaro tutte le quantita` e parametri che compaiono nella formula.

c. Scrivere lo pseudocodice di un algoritmo **ricorsivo efficiente** che calcola il valore della soluzione ottima per il problema illustrato. Si analizzi il tempo di esecuzione dell'algoritmo fornendo una stima asintotica quanto migliore e` possibile del tempo di esecuzione. **Si giustifichi in modo chiaro la risposta.**

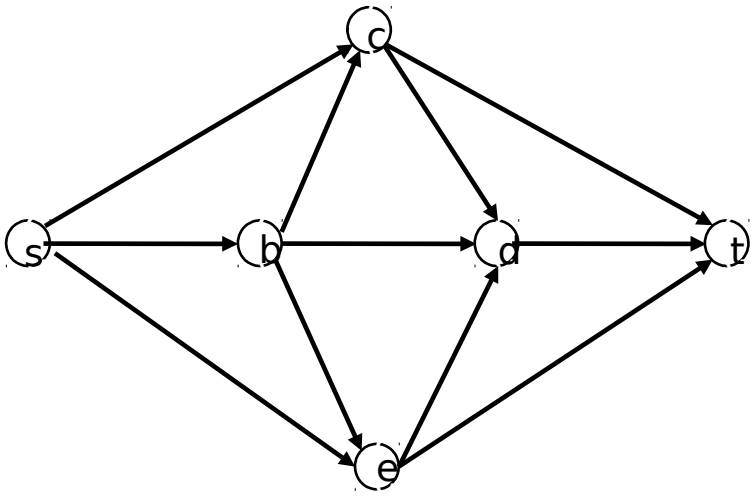
4. **Massimo flusso**
   a. Si consideri la seguente rete di flusso e la funzione di flusso i cui valori sono indicati a sinistra delle capacita` degli archi. Si disegni la rete residua rispetto alla funzione flusso indicata e si dica se questa funzione ha valore massimo. Nel caso in cui la funzione non abbia valore massimo, si fornisca la funzione flusso con valore massimo applicando una o piu` iterazioni dell'algoritmo di Ford-Fulkerson **a partire dalla funzione di flusso data**.

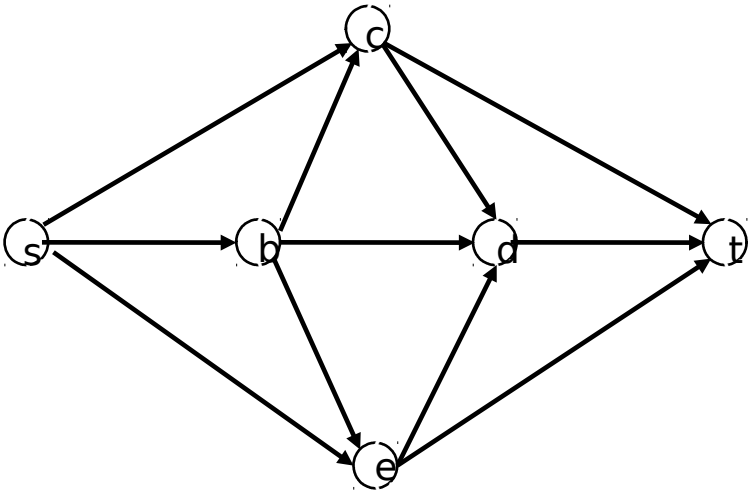
   Per ogni iterazione dell'algoritmo, occorre
   1. **disegnare la rete residua all'inizio di quell'iterazione**
   2. **indicare il cammino aumentante da voi scelto**
   3. **mostrare il valore associato ad ogni arco del grafo al termine di quella iterazione**

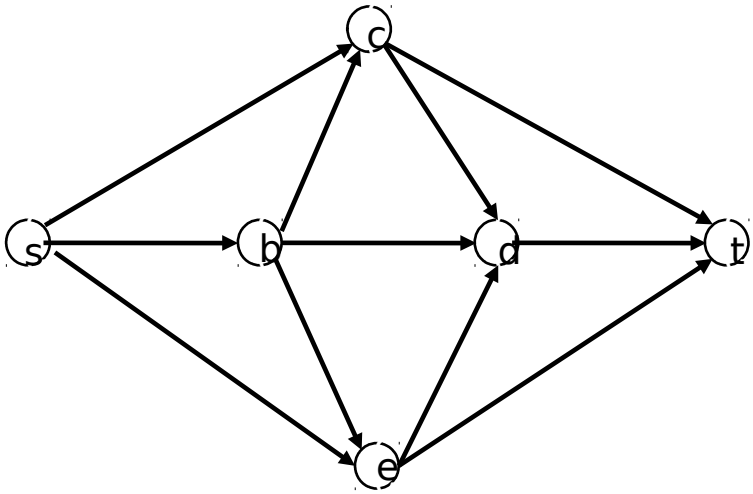
   N.B.: le risposte che non sono ottenute a partire dalla funzione di flusso data non saranno valutate.

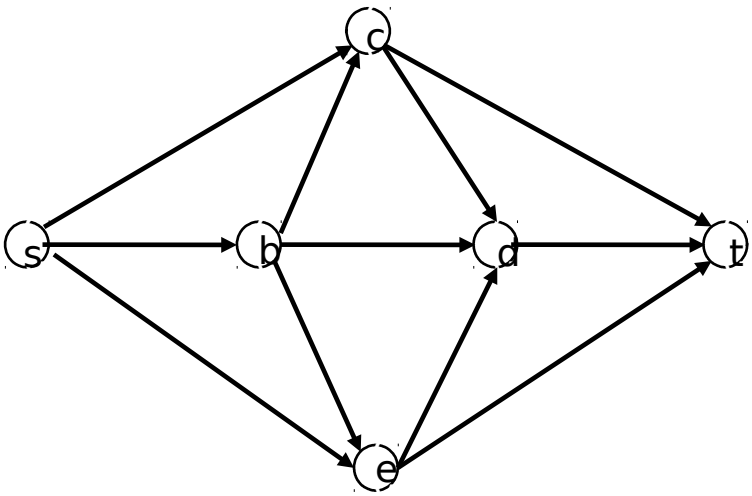


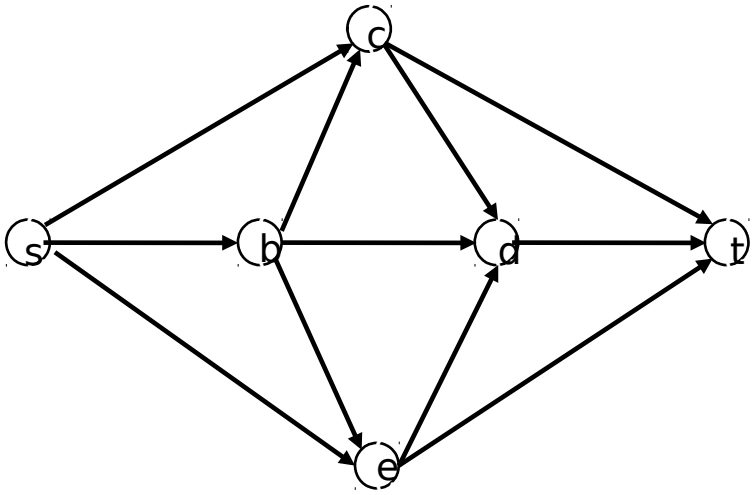
Per vostra comodita`, di seguito sono riportate diverse copie della rete di flusso, suddivise a coppie. **A partire dalla funzione di flusso data, usate l'immagine di sinistra di ciascuna coppia per disegnare la rete residua e l'immagine di destra per riportare i valori della funzione flusso assegnati a ciascun arco.** Ovviamente potrebbe essere necessario aggiungere e/o cancellare (con una x) archi nelle immagini di sinistra. Il numero di coppie non e` indicativo del numero di iterazioni effettuate dall'algoritmo di Ford-Fulkerson. Procedete dall'alto verso il basso utilizzando solo le coppie di grafi che vi servono per illustrare l'intera esecuzione dell'algoritmo.

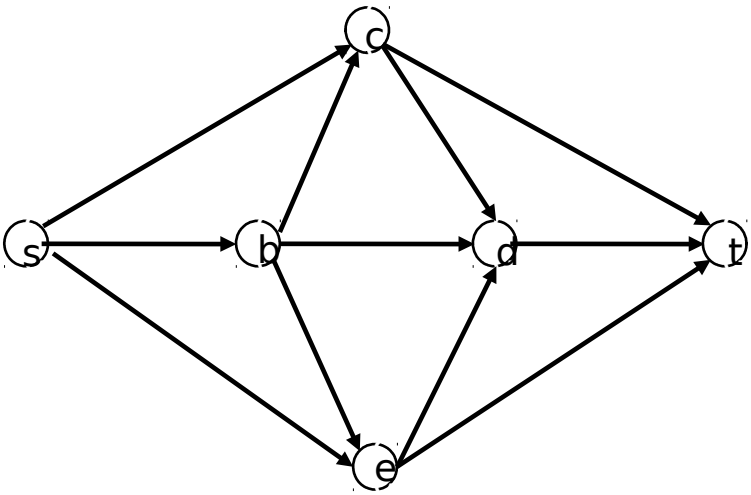

c
s
b
d
t
e

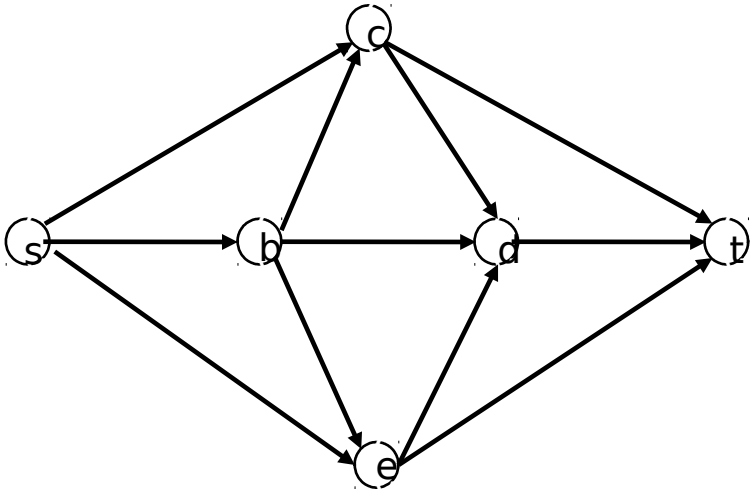

c
s
b
d
t
e

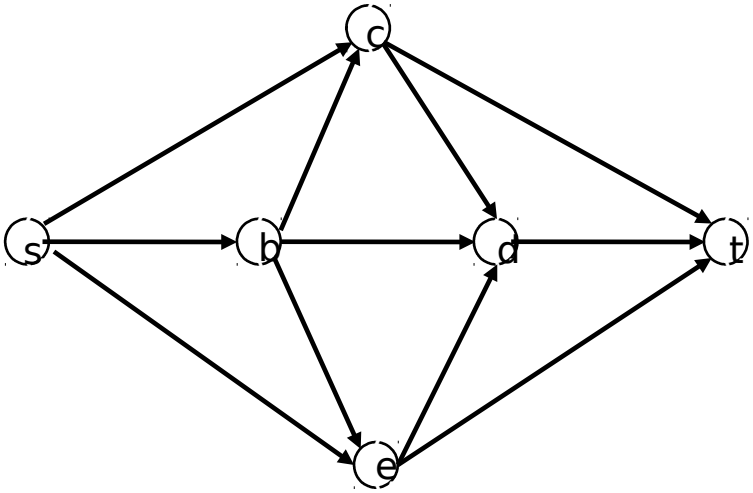

c
s
b
d
t
e

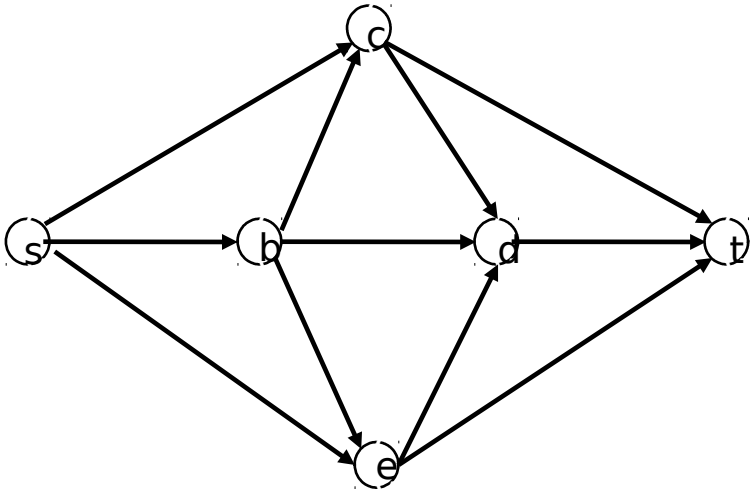

c
s
b
d
t
e

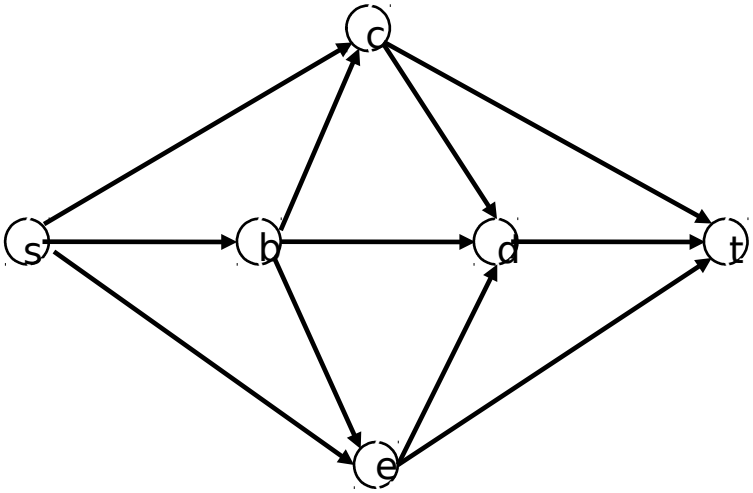

c
s
b
d
t
e

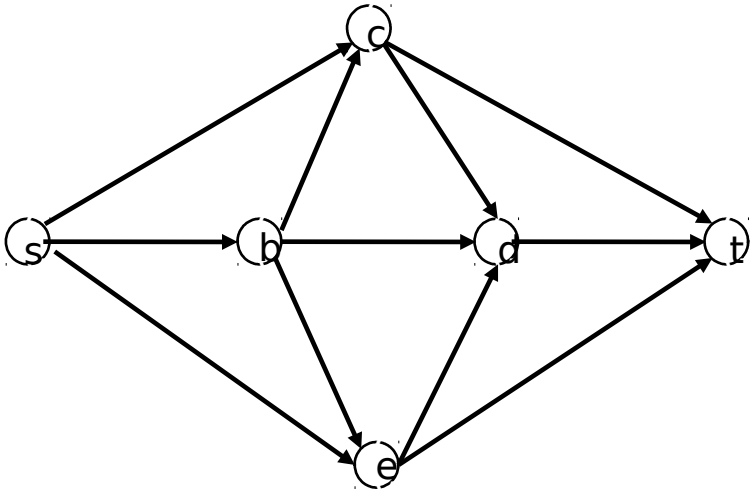

c
s
b
d
t
e

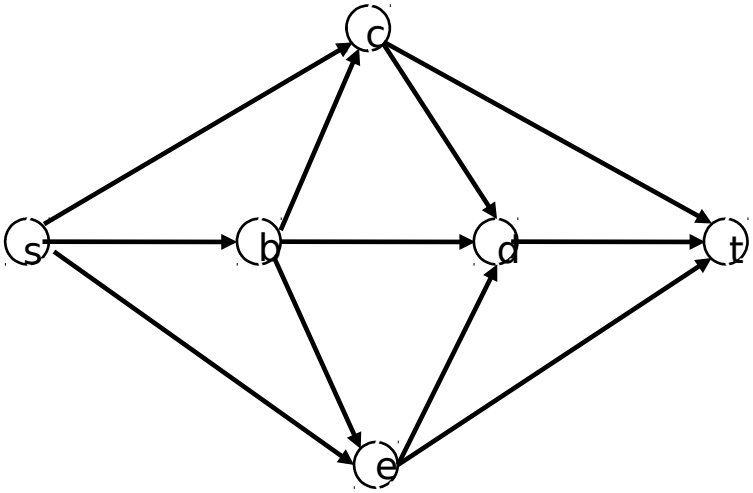

c
s
b
d
t
e


c
s
b
d
t
e


c
s
b
d
t
e


c
s
b
d
t
e


c
s
b
d
t
e


c
s
b
d
t
e

b. Descrivere in modo chiaro il comportamento dell'algoritmo che computa il massimo numero di percorsi disgiunti tra due dati nodi di un grafo direzionato. Occorre anche descrivere in che modo l'algoritmo alla fine costruisce i percorsi disgiunti.

c. Analizzare il tempo di esecuzione dell'algoritmo fornendo una stima asintotica quanto migliore e` possibile del tempo di esecuzione dell'algoritmo nel caso pessimo. **Giustificare in modo chiaro la risposta analizzando le singole fasi di cui consiste l'algoritmo.**

d. Dimostrare che il massimo numero di percorsi disgiunti tra due dati nodi e` minore o uguale del valore determinato dal vostro algoritmo (ovviamente vale l'uguaglianza ma qui e` sufficiente dimostrare il <=)