Progettazione di Algoritmi (9 CFU)
28/6/2018

Cognome e Nome:
Numero di Matricola:

Spazio riservato alla correzione

1	2	3	4	5	6	Totale
/20	/18	/15	17	15	15	/100

1. Analisi degli algoritmi e notazione asintotica

a) Indicare quali delle seguenti affermazioni sono vere e quali sono false.
   1. $(\log n)^2+100 \log n+1= O(n \log n)$
   2. $2^{10} = \Omega(n^{1/4})$
   3. $n= O(\log n)$
   4. $n^3-1000n^2+8=\Omega(n^3)$
   5. $¼\ n = \Omega(n)$

b) Si dimostri che se $0<f(n)=O(h(n))$ e $0<g(n)=\Omega(p(n))$ allora $f(n)/g(n)=O(h(n)/p(n))$.
Occorre utilizzare solo la definizione di O e di Ω e nessuna altra proprieta`.

c) Si analizzi il tempo di esecuzione nel caso pessimo del seguente segmento di codice fornendo una stima asintotica **quanto migliore e` possibile** per esso. **Si giustifichi in modo chiaro la risposta.**

```
FOR(i=1; i≤n; i=i+1){

      FOR(k=0; k<10; k=k+1) {
       print(k);
      }

      FOR(j=1; j<i; j=j*2) {
       print(j);
    }


  }
```

2. Divide et Impera

a) Si scriva lo pseudocodice dell'algoritmo QuickSelect e dell'algoritmo Distribuzione.

b) Si fornisca la relazione di ricorrenza che esprime un limite superiore al tempo di esecuzione dell'algoritmo QuickSelect per un generico rango r_p del pivot. Si spieghi in modo chiaro come si arriva a formulare la relazione di ricorrenza da voi fornita.

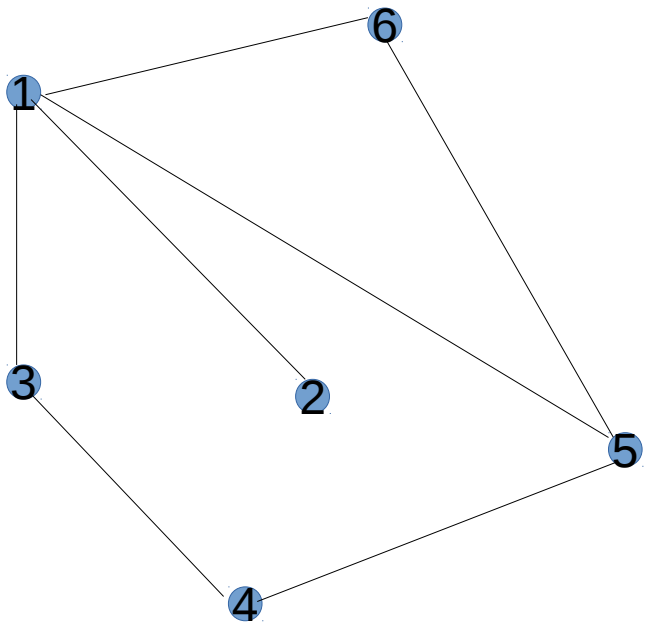
c) A partire dalla relazione di ricorrenza fornita al punto precedente, **si calcoli** una stima asintotica quanto migliore e` possibile del tempo di esecuzione dell'algoritmo QuickSelect nel caso pessimo. Non e` sufficiente fornire la stima asintotica ma e` necessario mostrare come essa viene ottenuta a partire dalla relazione di ricorrenza.

Grafi

a) Spiegare in cosa consiste l'ordinamento topologico di un grafo direzionato aciclico.

b) Si scriva lo pseudocodice di un algoritmo **ricorsivo** che computa l'ordinamento topologico di un grafo direzionato aciclico in tempo O(n+m). Si descrivano tutte le strutture dati utilizzate dall'algoritmo e si dimostri che l'algoritmo ha tempo di esecuzione O(n+m) nel caso pessimo.

c) Disegnare l'albero BFS che si ottiene eseguendo una visita BFS sul seguente grafo a partire dal nodo 1. **Si assuma che i nodi siano disposti nelle liste di adiacenza in ordine crescente di etichetta.**



3. Algoritmi greedy

a) Si spieghi in che cosa consiste un'istanza (input) del problema del caching offline e in cosa consiste una soluzione (output) del problema.

b) Si spieghi che cosa e` un eviction scheduling ridotto (se non lo si e` gia spiegato al punto precedente) e si dimostri che e` sempre possibile trasformare un eviction schedule in un eviction schedule ridotto senza aumentare il numero totale di inserimenti nella cache.

c) Si illustri in che modo puo` essere usata una coda a priorita` per implementare l’algoritmo di Belady. Non e` necessario fornire lo pseudocodice dell’algoritmo.

4. Programmazione dinamica

a) Si fornisca una formula ricorsiva per computare il valore OPT(i,v) della soluzione ottima per il problema dei cammini minimi. Si spieghi **in modo chiaro** come si arriva a questa formula. **Prima di procedere pero` si completi la seguente frase. L'esercizio 3 sara` valutato solo se lo studente fornira` una definizione corretta di OPT(i,v).**

OPT(i,v) = lunghezza del …

b) In generale, per quanti valori di i dobbiamo computare OPT(i,v) per essere certi di ottenere il valore piu` piccolo di OPT(i,v) tra tutti i possibili valori di i? **Giustificare la risposta.**

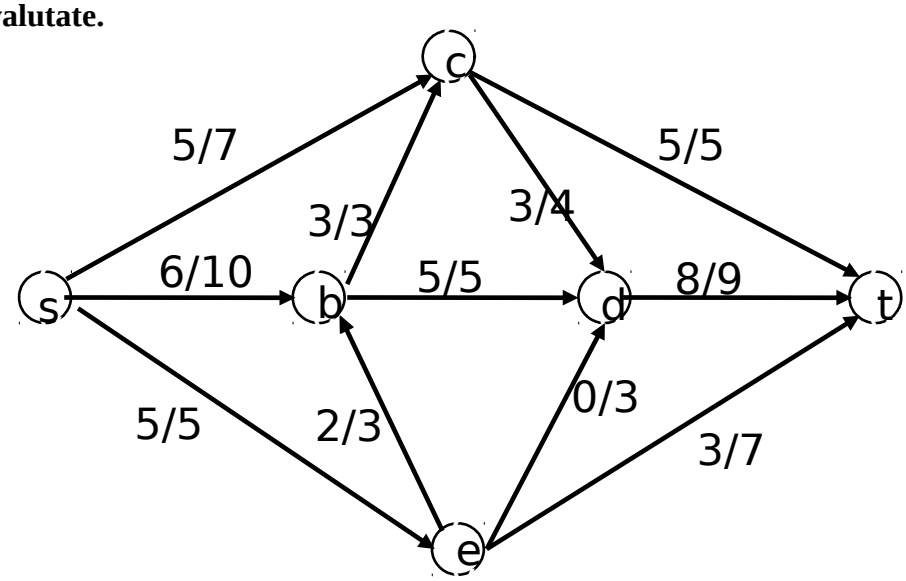
4. **Massimo flusso**

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 e il taglio di capacita` minima.** A tal fine si eseguano una o piu` iterazioni dell'algoritmo di Ford-Fulkerson **a partire dalla funzione di flusso data.**

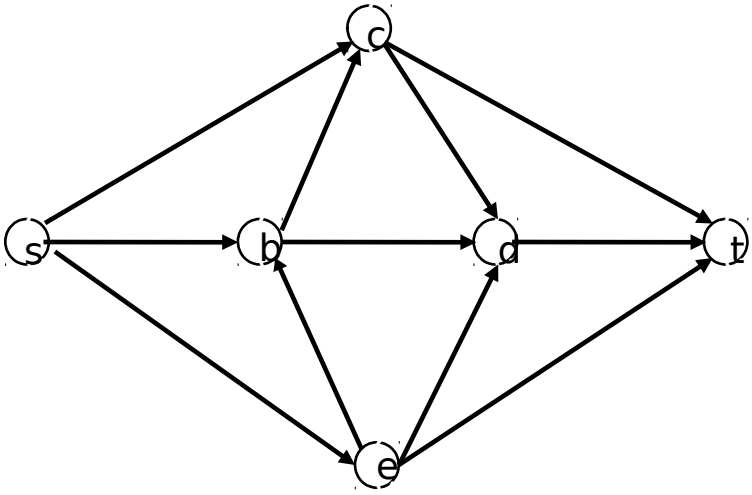
Per ogni iterazione dell'algoritmo, occorre

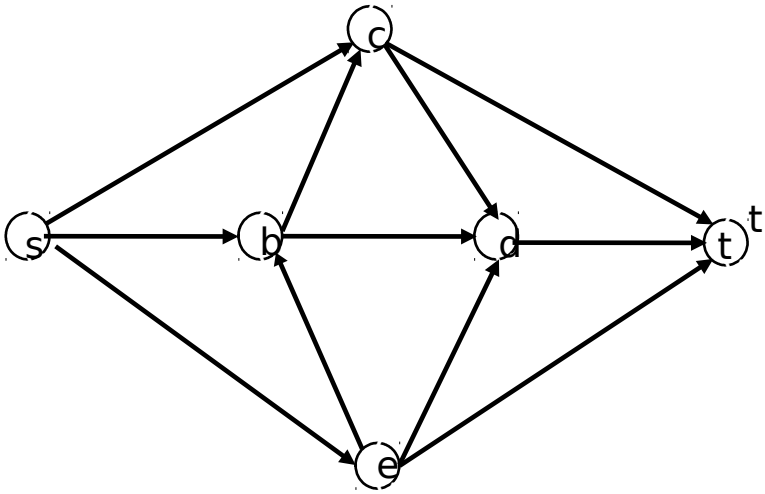
- **disegnare la rete residua all'inizio di quell'iterazione**
- **indicare il cammino aumentante da voi scelto**
- **mostrare il valore associato ad ogni arco del grafo al termine di quella iterazione**

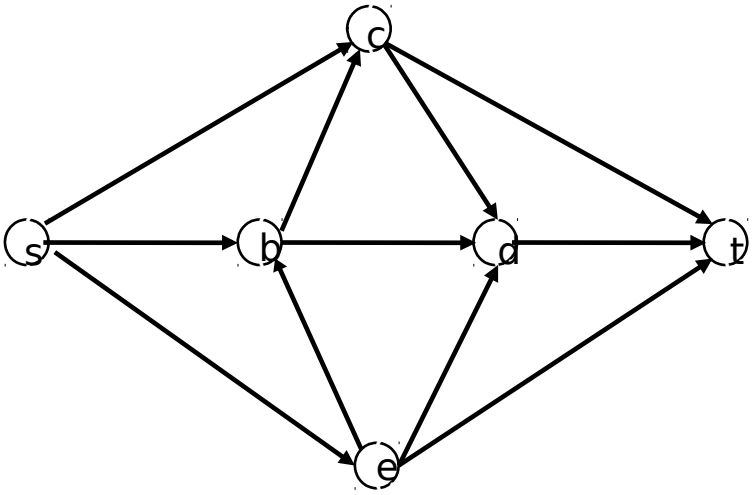
N.B.: le risposte che non sono ottenute a partire dalla funzione di flusso data non saranno valutate.

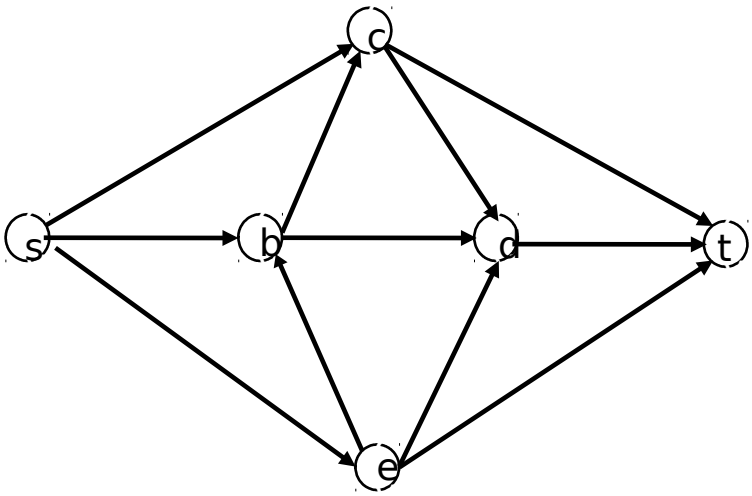


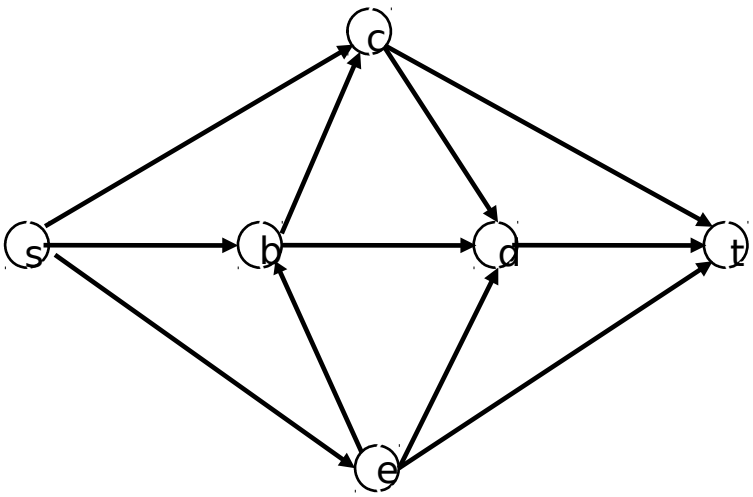
Per vostra comodita`, di seguito sono riportate diverse copie della rete di flusso, suddivise a coppie. **A partire dalla funzione di flusso data, usate l'immagine di sinistra di ciascuna coppia per disegnare la rete residua e l'immagine di destra per riportare i valori della funzione flusso assegnati a ciascun arco.** Ovviamente potrebbe essere necessario aggiungere e/o cancellare (con una x) degli 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. **N.B.: Se non saranno rispettate queste indicazioni per lo svolgimento dell'esercizio, l'esercizio non sara` valutato.**

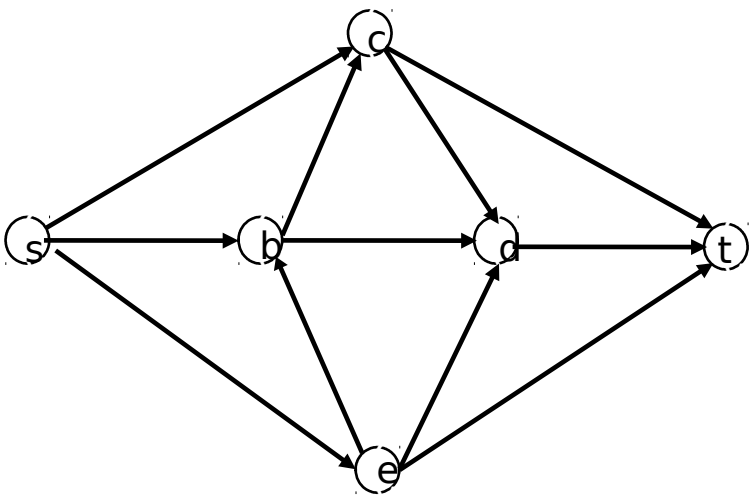

c
s
b
d
t
e

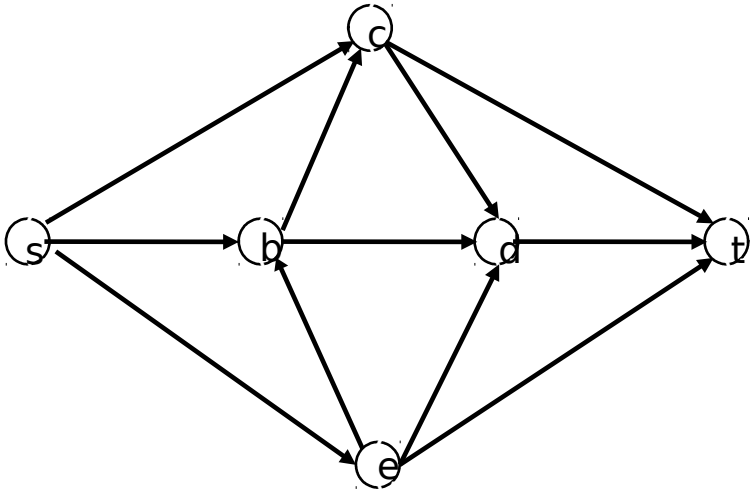

c
s
b
d
t
e

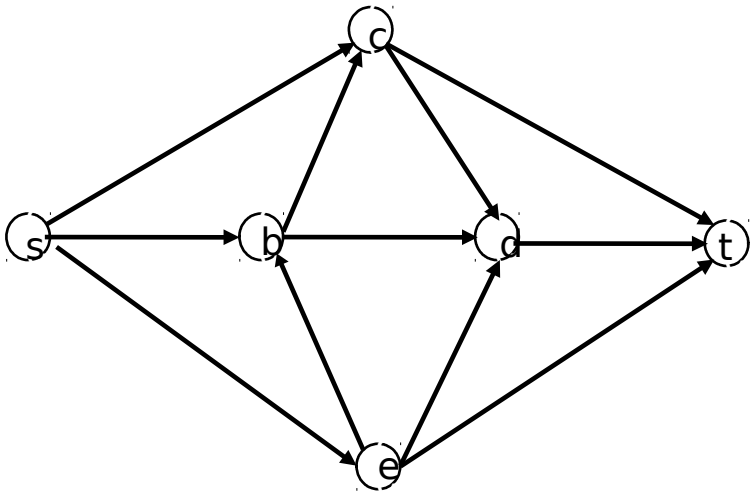

c
s
b
d
t
e

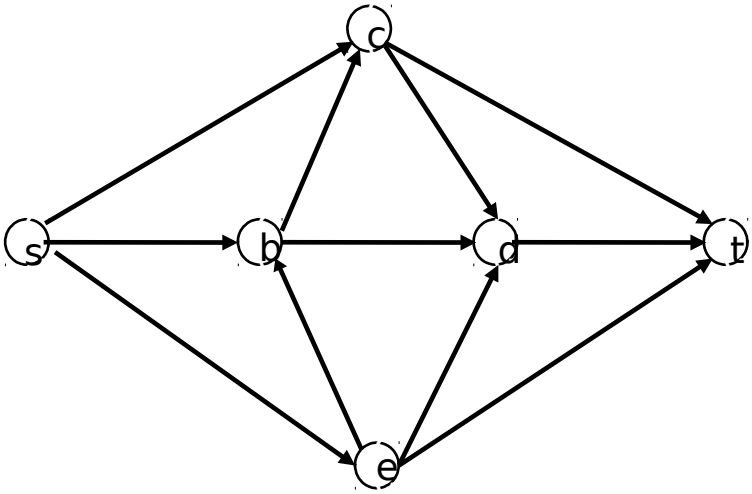

c
s
b
d
t
e

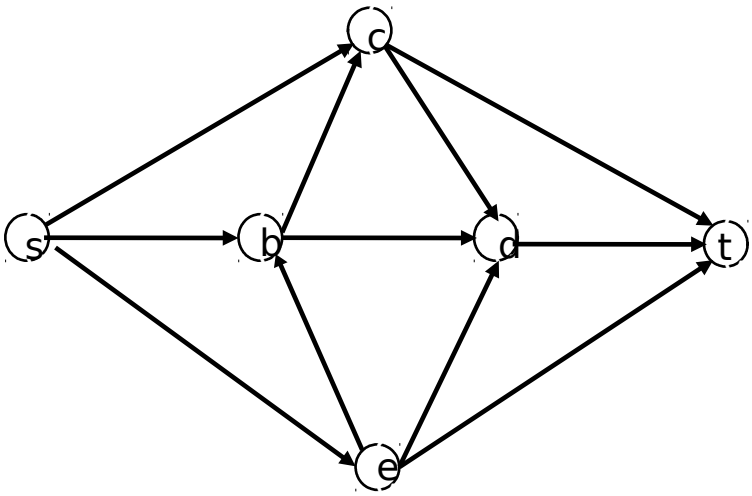

c
s
b
d
t
e

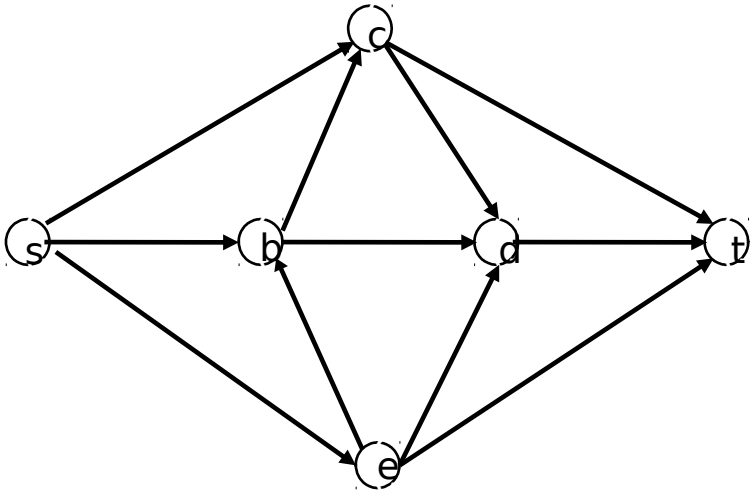

c
s
b
d
t
e

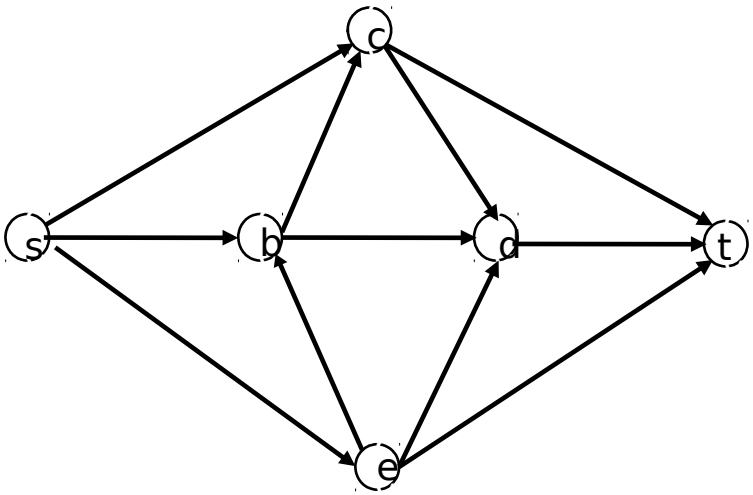

c
s
b
d
t
e


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) Scrivere lo pseudocodice dell'algoritmo di Ford-Fulkerson e dell'algoritmo Augment da esso invocato.