

Cognome e Nome:
Numero di Matricola:

Spazio riservato alla correzione

| 1 | 2 | 3 | 4 | 5 | 6 | Totale |
|-----|-----|-----|-----|----|----|--------|
| /18 | /15 | /20 | /20 | 13 | 14 | /100 |

1.

a) Indicare quali delle seguenti affermazioni sono vere e quali sono false.

- $\log n + n^3 - n^2 = O(n^3)$
- $\sqrt{n} = \Omega(\log^2 n)$
- $n^{1/2} = O(n^{1/2})$
- $n^3 + 10^4 n^2 + 8 = O(n^3)$
- $16^{\log_4 n} = O(n)$

b) Si dimostri in modo formale e facendo uso direttamente della definizione di O e di Ω (**senza ricorrere ad altre proprietà**) che se $f(n) = O(g(n))$ e $h(n) = \Omega(g(n))$ allora $f(n) = O(h(n))$

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

```
FOR(i=1; i≤4n; i=i*4){  
    FOR(k=1; k<1000; k=2*k) {  
        print(k);  
    }  
    print(i);  
}
```

2.

- a) Si scriva lo pseudocodice dell'algoritmo MergeSort e dell'algoritmo Fusione.

- b) Si fornisca la relazione di ricorrenza che esprime il tempo di esecuzione dell'algoritmo MergeSort e si dica qual è il tempo di esecuzione dell'algoritmo nel caso pessimo (non occorre fornire alcuna dimostrazione).

3. Grafi

a) Fornire lo pseudocodice un algoritmo ricorsivo che in $O(n+m)$ trova l'ordinamento topologico di un DAG. **Occorre aggiungere allo pseudocodice anche le istruzioni che consentono di ottenere il tempo di esecuzione $O(n+m)$ illustrando il significato delle strutture dati utilizzate dall'algoritmo. Se non si è in grado di aggiungere queste istruzioni (pseudocodice NON descrizione), le si ometta tenendo conto che in questo caso si perderanno dei punti.**

b) Analizzare il tempo di esecuzione nel caso peggior caso dell'algoritmo di cui al punto precedente. Se il vostro algoritmo non ha tempo di esecuzione $O(n+m)$ anche a questo punto sarà attribuito un numero inferiore di punti.

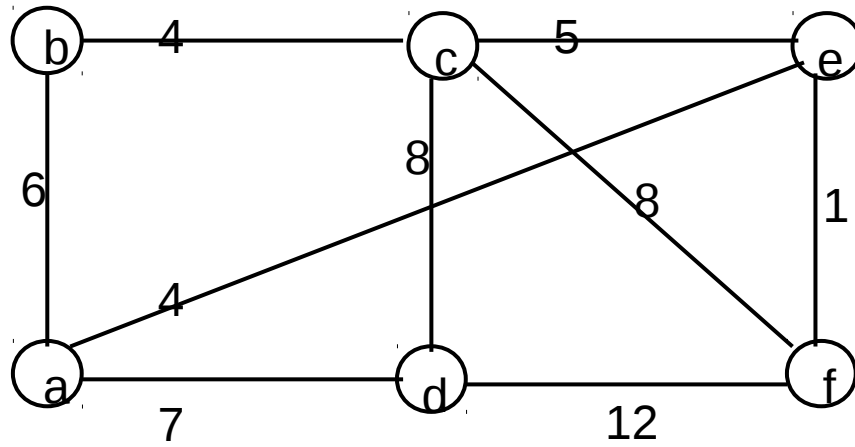
c) Dimostrare che in un DAG c'è almeno un nodo senza archi entranti.

4. Algoritmi greedy

a) Si scriva lo **pseudocodice** dell'algoritmo di Prim che fa uso della coda a priorità **con l'aggiunta delle istruzioni che consentono di costruire il minimo albero ricoprente. Si illustri il significato delle eventuali strutture dati utilizzate da queste istruzioni aggiuntive. Se non si è in grado di aggiungere queste istruzioni (pseudocodice NON descrizione), le si ometta tenendo conto che in questo caso si perderanno dei punti.**

b) Dire in quale passo dell'algoritmo viene effettuata la scelta greedy e perché ogni volta questa scelta preserva la proprietà che l'albero costruito fino a quel punto è un sottoinsieme dello MST?

c) Si mostri l'esecuzione dell'algoritmo di Kruskal sul grafo riportato di seguito. Per ciascun passo, occorre disegnare la foresta corrispondente a quel passo.



5. Programmazione dinamica

Si consideri la seguente formula, dove b è un numero reale maggiore o uguale di 0 e j è un intero maggiore o uguale di 0.

$$\text{OPT}(0,b)=0$$

$$\text{OPT}(j,b)=\text{OPT}(j-1,b) \text{ se } b < b_j$$

$$\text{OPT}(j,b)=\max\{\text{OPT}(j-1,b), \text{OPT}(j-1,b-b_j)+b_j\} \text{ altrimenti}$$

Ciascuno dei seguenti punti sarà valutato solo se si è risposto in modo corretto ai punti che lo precedono.

- a) Si descriva un problema di ottimizzazione per il quale la suddetta formula fornisce il valore della soluzione ottima. **Non è sufficiente fornire il nome del problema ma occorre descrivere l'input e l'output del problema.**

b) Si spieghi in modo chiaro cosa rappresenta $OPT(j,b)$

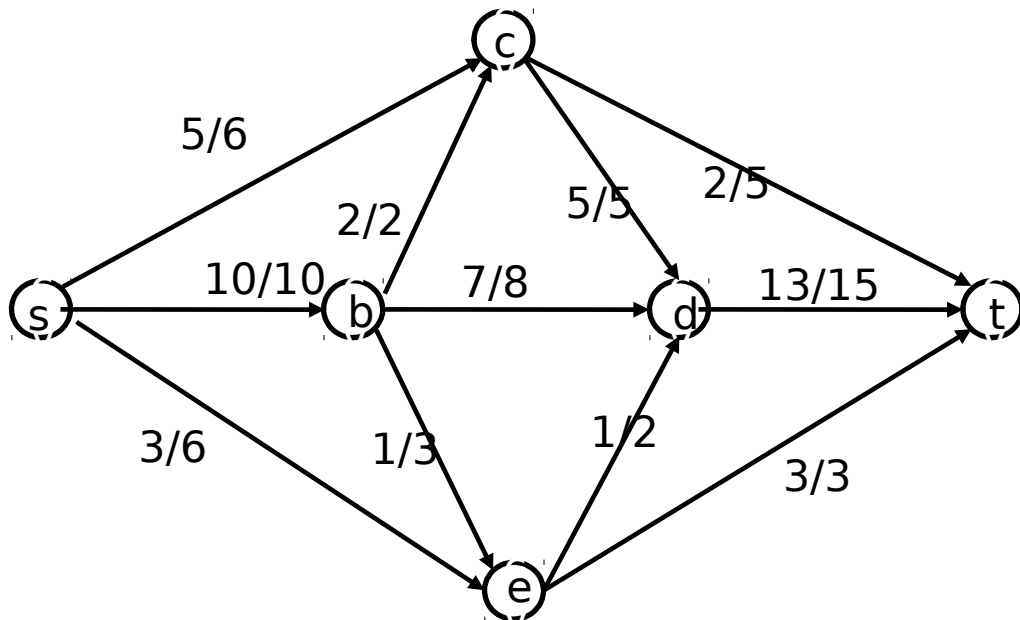
c) Si spieghi in modo chiaro come si arriva alla formula di cui al punto a).

6. Massimo flusso

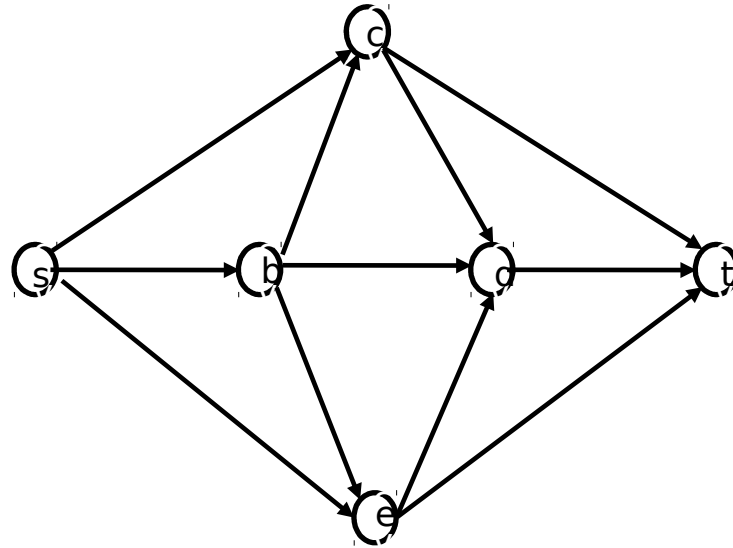
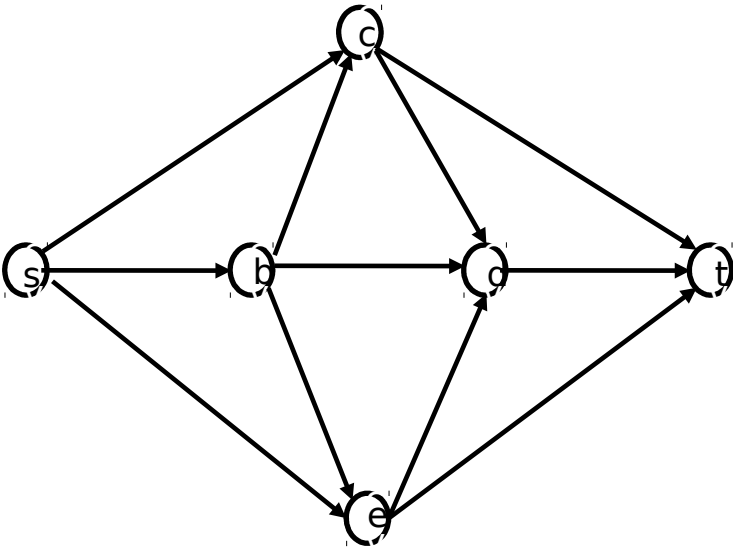
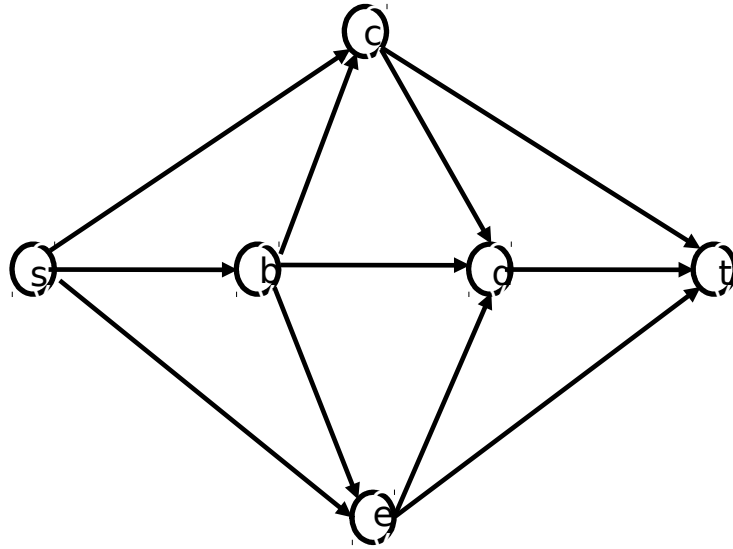
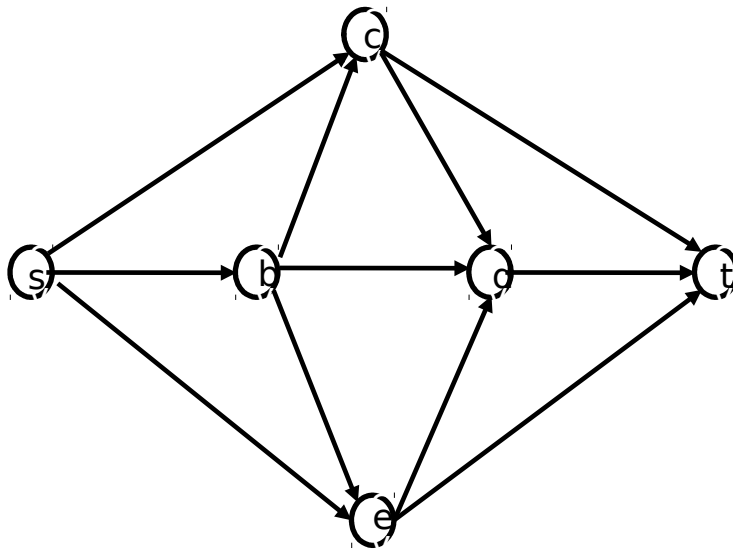
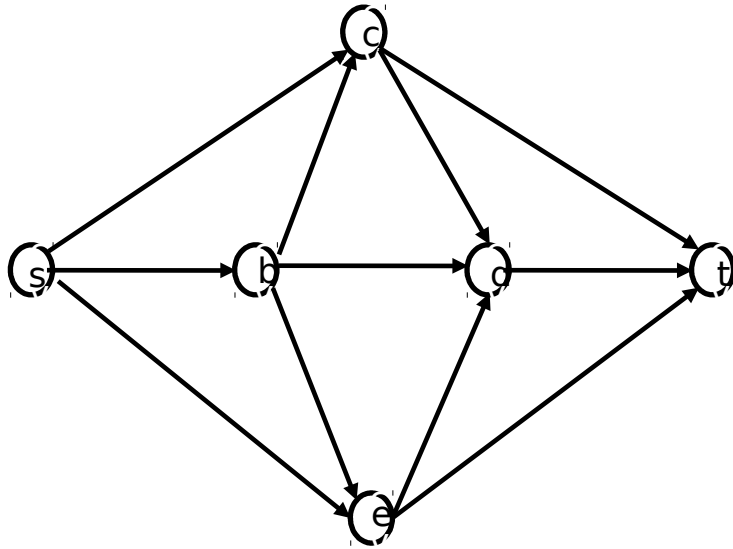
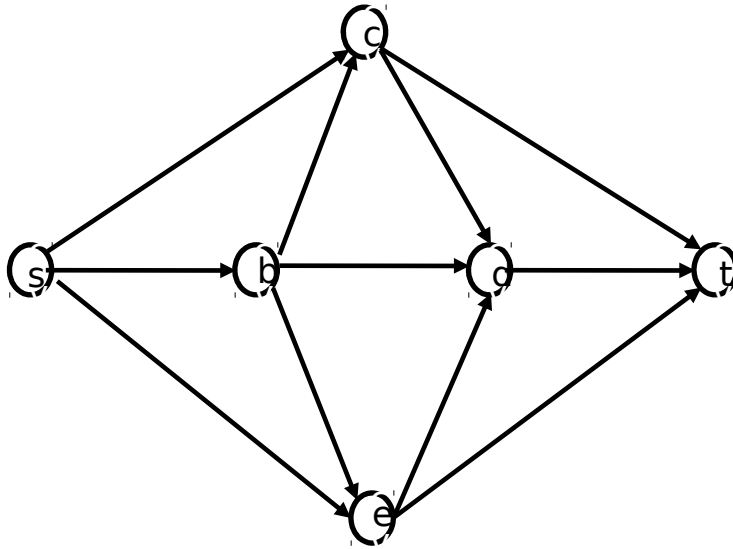
a) Si consideri la seguente rete di flusso e la funzione di flusso i cui valori sono indicati a sinistra delle capacità degli archi.

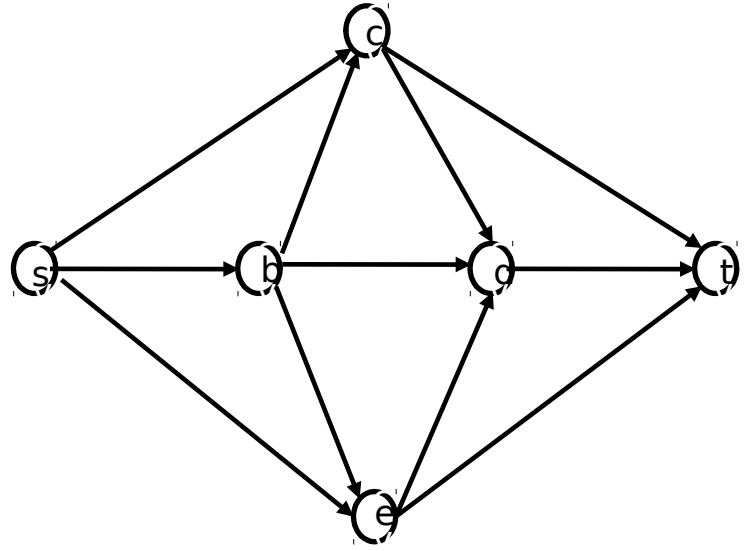
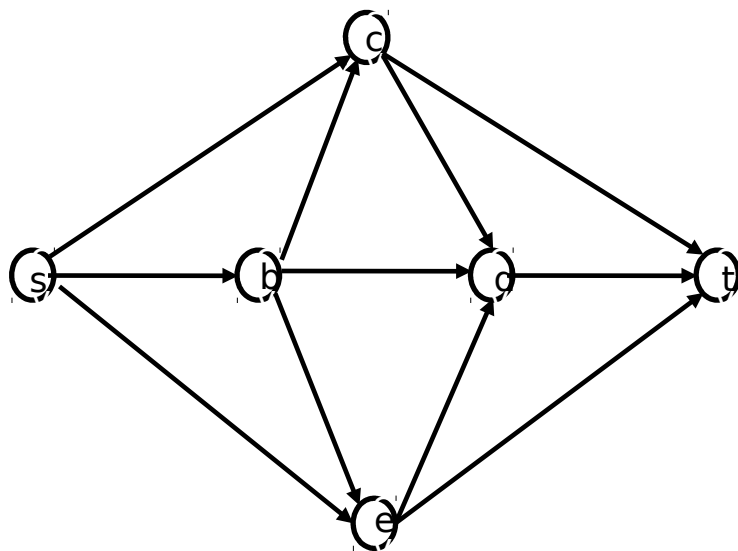
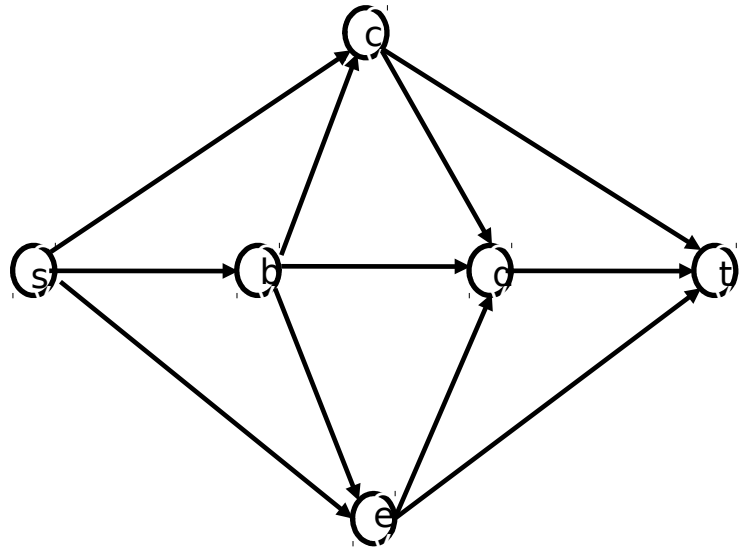
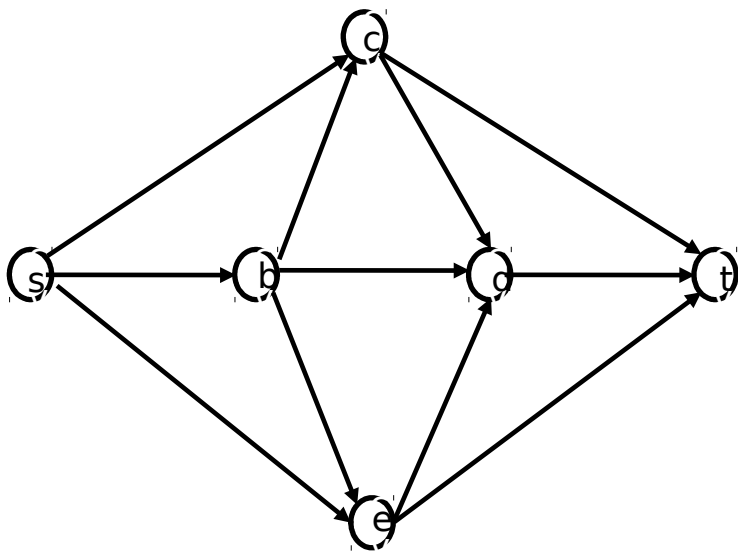
- Si disegni la rete residua rispetto alla funzione flusso indicata e **si dica se questa funzione ha valore massimo**.
- Nel caso in cui la funzione non abbia valore massimo, si fornisca la funzione flusso con valore massimo applicando l'algoritmo di Ford-Fulkerson **a partire dalla funzione di flusso data**. Per ogni iterazione dell'algoritmo, occorre disegnare la rete residua all'inizio di quell'iterazione, indicare il cammino aumentante scelto e mostrare il flusso associato ad ogni arco della rete di flusso originaria al termine di quella iterazione
- Si dica qual è il **valore del massimo flusso** e si fornisca un **taglio di capacità minima**.

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



Per vostra comodità, 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 cancellare (con una x) archi nelle immagini di sinistra. Il numero di coppie di reti non è 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.





b) Si dimostri che se non esistono cammini aumentanti nella rete residua di G rispetto al flusso f allora f è massimo.