

NOME:

COGNOME:

MATRICOLA:

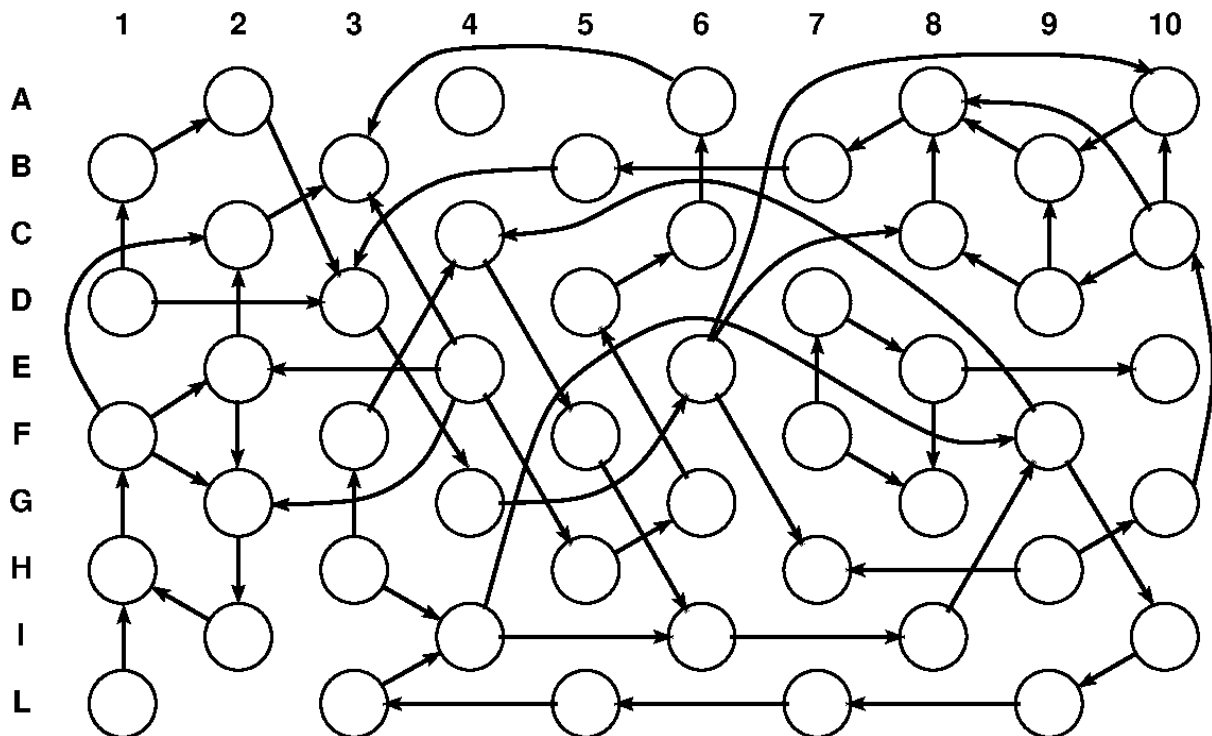
FIRMA:

Esame di Ricerca Operativa - 25 settembre 2025

Facoltà di Scienze MM.FF.NN. - Verona

4 esercizi per 101 punti in palio voto \geq punti - 5, 40 \rightarrow 30 e lode)

Esercizio 1 (con 7 richieste: 1+1+3+1+1+1+5 = 13 punti [grafi visual]):



Tranne che per l'ultima richiesta, si ignorino le direzioni degli archi in figura. Eventuali nozioni mancanti possono essere pubblicamente richieste al docente.

Richieste dell'Esercizio 1

1.1 (1 pt, componenti_connesse) Colora i nodi in modo da evidenziare le diverse componenti connesse

1.2 (1 pt, distingui nodi e archi speciali)

nodì isolati	foglie	cutnodes	bridges

1.3 (3 pt, make bipartite) rendi il grafo bipartito rimuovendo il minor numero di archi (1pt se suggerisci quali archi rimuovere ed evidenzi la bipartizione del grafo risultante, 1pt se esibisci una famiglia di cicli dispari che richiedano la rimozione di quel numero di archi per essere tutti eliminati, 1pt per il numero di soluzioni ottime). Addobba sempre la figura sopra per l'esibizione dei certificati

1.4 (1 pt, planarità) Dire se l'intero grafo è planare oppure no, argomentandolo via certificati

1.5 (1 pt, Hamilton) Per ogni componente di più nodi, fornire un ciclo Hamiltoniano se presente, altrimenti un cammino Hamiltoniano se presente, altrimenti spiega perchè no

1.6 (1 pt, Eulero) Per ogni componente di più nodi, fornire un ciclo Euleriano se presente, altrimenti

un cammino Euleriano se presente, altrimenti spiega perchè no

1.7 (5 pt, strong-connectivity) Si riguardino ora gli archi come diretti, ciascuno orientato come in figura. Per ciascuna componente tranne il nodo isolato si fornisca un ordinamento topologico oppure un ciclo diretto in essa contenuto (1pt). Per ogni componente non-aciclica si evidenzino: le componenti fortemente connesse (1pt), il DAG delle componenti fortemente connesse (1pt) e il relativo ordine topologico (1pt) e si certifichi la forte connessione di ciascuna componente offrendone costruzione per aggiunta di orecchie partendo da un ciclo diretto (1pt)

Esercizio 2 (con 14 richieste: 1+1+1+1+1+1+6+2+1+2+1+6+5+10 = 39 punti [modellazione/riduzioni]):

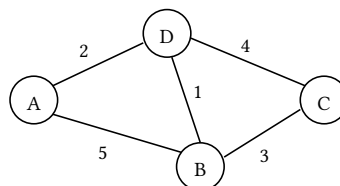
Il modello di PL qui a destra (in alto) computa un albero ricoprente di peso minimo (MST, minimum spanning tree) in un generico grafo $G = (V, E)$ con costi non-negativi sugli archi $c : E \rightarrow \mathbb{R}_+$. Il modello coinvolge un vettore x di variabili reali, indicizzato dal set E . Per ogni $F \subseteq E$, definiamo $x(F) := \sum_{e \in F} x_e$. Dove \mathbb{S} è l'insieme dei sottoinsiemi di V non-banali (ossia non vuoti e non uguali a V), per ogni $S \in \mathbb{S}$ si indica con $\delta(S)$ l'insieme degli archi con un estremo in S e l'altro in $V \setminus S$ (ossia gli archi del taglio di spiagge S e $V \setminus S$).

$$\min x(E)$$

$$x(\delta(S)) \geq 1 \text{ for all } S \in \mathbb{S}$$

$$x \geq 0$$

your instance:



Richieste dell'Esercizio 2

- 2.1 (1 pt, one MST)** dare un MST per il grafo in figura
- 2.2 (1 pt, unicity)** Certifica l'unicità dell'MST avvalendoti del lemma dei tagli. Quali tagli hai dovuto osservare/esibire?
- 2.3 (1 pt, explicit instance-specific)** Scrivi esplicitamente il modello di PL che computa l'MST per l'istanza assegnata in figura. (Istanza cioè il modello generale fornito per l'istanza specifica.)
- 2.4 (1 pt, canonic form)** In che forma particolare è il problema di PL che hai appena scritto? Perché? Mettilo in forma canonica definendo le variabili di slack.
- 2.5 (1 pt, auxiliary problem)** Poichè il problema non è ad origine ammissibile, si consideri il problema ausiliario.
- 2.6 (1 pt, first tableau)** Scrivi il primo tableau del problema ausiliario.
- 2.7 (6 pt, first phase)** Risolvi all'ottimo il problema ausiliario col metodo del simplesso (1 punto per ogni pivot e 1 punto per ogni prova del nove esplicitamente spesa e chiaramente rappresentata sul foglio).
- 2.8 (2 pt, change horse)** Ottieni un tableau del problema originale che esprima una soluzione di base ammissibile. Un punto è per il recupero della funzione obiettivo originaria.
- 2.9 (1 pt, second phase)** Ottieni il tableau che esprime la soluzione ottima del problema originale.
- 2.10 (2 pt, dual)** Recupera dall'ultimo tableau le soluzioni primale e duale ottime (1pt) ed interpretale (1pt) entrambe nel contesto del problema di MST che intendevi risolvere sull'istanza assegnata.
- 2.11 (1 pt, dual opts)** La soluzione duale è unica? Sì/no e perchè.
- 2.12 (6 pt, complementary slackness)** Si indichi ora col parametro K il costo dell'arco AD . La soluzione ottima ottenuta sopra rimarrà comunque ammissibile. Poniti la sfida di stabilire se sia ottima o meno (dipenderà dal valore di K) tramite le condizioni degli scarti complementari. Riscrivi il problema

primale e duale introducendo il parametro K (1pt). Imposta le condizioni degli scarti complementari relative alla nostra soluzione primale (2pt). Ottieni soluzione duale complementare ad essa (1pt). Come puoi ora stabilire il range di valori di K per cui le due soluzioni sono ottime (2pt)?

2.13 (5 pt, Steiner tree problem) il problema MST è un caso particolare dello Steiner tree problem dove oltre al grafo $G = (V, E)$ con costi sugli archi viene dato in input un sottoinsieme T di V e all'albero da ritornare non si richiede più di raggiungere tutti i nodi in V ma basta che raggiunga tutti i nodi di T (può raggiungere anche nodi in $V \setminus T$ ma solo nella misura in cui ciò aiuti nel contenere il costo, o quantomeno non lo aumenti). Fornire formulazione di Programmazione Lineare Intera (PLI) dello Steiner tree problem (2pt) prendendo ispirazione dalla formulazione di PL per l'MST fornita con l'esercizio (consentiamo quindi che il numero di vincoli sia esponenziale nelle dimensioni dell'istanza). Dare poi però un esempio dove la soluzione ottima del problema di PL rilassato (ottenuto ignorando i vincoli di interezza) è frazionaria (3pt).

2.14 (10 pt, NPC-proof) Dimostrare l'NP-hardness dello Steiner tree problem con una riduzione da MIN NODE COVER, il problema di rimuovere il minimo numero di nodi dal grafo per lasciarlo privo di archi (5pt). Altri 5pt sono in palio per una riduzione da 3-SAT.

Esercizio 3 (con 12 richieste: 1+1+1+1+1+1+1+1+2+1+1+2 = 14 punti [programmazione dinamica]):

La seguente tabella offre, nella sua terza riga, una sequenza S di numeri naturali (la prima riga, a caratteri in neretto, serve solo ad indicizzarla).

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
31	10	19	50	60	17	28	39	45	26	36	59	21	42	30	55	57	20	22	14	37	56	24	47	57	40	23

Richieste dell'Esercizio 3

3.1 (1 pt, DP: last_in_pos) Alla tabella si aggiunga una riga che in ogni posizione i , con $1 \leq i \leq n$, riporti la massima lunghezza di una sottosequenza strettamente crescente di S che di S prenda l'elemento in posizione i come suo ultimo elemento.

3.2 (1 pt, DP: first_in_pos) Si aggiunga una riga che in ogni posizione i , con $1 \leq i \leq n$, riporti la massima lunghezza di una sottosequenza strettamente crescente di S che di S prenda l'elemento in posizione i come suo primo elemento.

3.3 (1 pt, opt_sol: libera) Trovare una sottosequenza strettamente crescente di S di massima lunghezza. Specificare quanto è lunga e fornirla.

3.4 (1 pt, certify opt) Fornire un minimo numero di sottosequenze mai crescenti tali che ogni elemento della sequenza originale in input ricada in almeno una di esse. Specificare quante sono e fornirle.

3.5 (1 pt, opt_sol: last) Trovare una sottosequenza strettamente crescente di S di massima lunghezza tra quelle che terminano con l'elemento in posizione 19. Specificare quanto è lunga e fornirla.

3.6 (1 pt, opt_sol: left) Trovare una sottosequenza strettamente crescente di S di massima lunghezza tra quelle che di S non prendono alcun elemento di indice inferiore a 8. Specificare quanto è lunga e fornirla.

3.7 (1 pt, opt_sol: prende) Trovare una sottosequenza strettamente crescente di S di massima lunghezza tra quelle che di S prendono l'elemento in posizione 13. Specificare quanto è lunga e fornirla.

3.8 (1 pt, N-sequenza) Una sequenza è detta una N-sequenza, o sequenza strettamente crescente con

al più un ripensamento, se esiste un indice i tale che ciascuno degli elementi della sequenza, esclusi al più il primo e l' i -esimo, è strettamente maggiore dell'elemento che lo precede. Trovare la più lunga N-sequenza che sia una sottosequenza della sequenza data. Specificare quanto è lunga e fornirla.

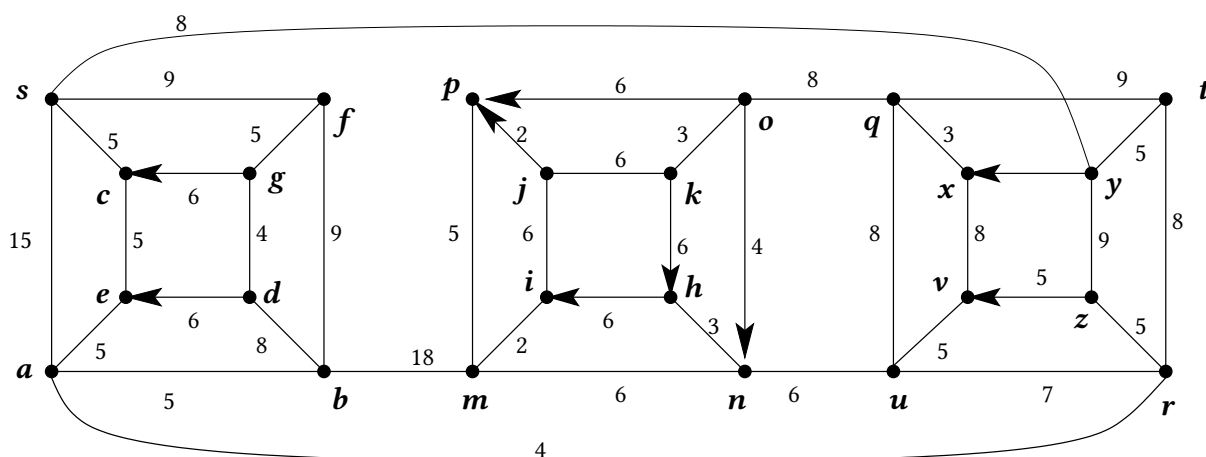
3.9 (2 pt, quante opt sol: libere) Le sottosequenze di S sono 2^n , in corrispondenza biunivoca coi sottoinsiemi degli indici degli elementi di S che includono. Stabilire quante siano le sottosequenze strettamente crescenti di S di massima lunghezza.

3.10 (1 pt, opt_sol: last) Quante sono le sottosequenze strettamente crescenti di S di massima lunghezza tra quelle che prendono l'elemento in posizione 19 come loro ultimo elemento?

3.11 (1 pt, opt_sol: first) Quante sono le sottosequenze strettamente crescenti di S di massima lunghezza tra quelle che prendono l'elemento in posizione 8 come loro primo elemento?

3.12 (2 pt, quante opt sol: con) Quante sono le sottosequenze strettamente crescenti di S di massima lunghezza tra quelle che di S includono l'elemento in posizione 13?

Esercizio 4 (con 11 richieste: $8+2+3+4+2+2+1+2+5+4+2 = 35$ punti [grafi]):



Richieste dell'Esercizio 4

- 4.1 (8 pt, recognize planarity) Dire, certificandolo, se siano planari o meno il grafo G , il grafo G_u ottenuto da G sostituendo l'arco sy con un arco su , e il grafo G_q ottenuto da G sostituendo l'arco sy con un arco sq (3pt). Per quelli di questi tre grafi che non siano planari, identifica gli archi la cui rimozione renderebbe il grafo planare (1pt) corredando ciascuno arco col relativo planar embedding (2pt) e fornendo certificati di non planarit  in quantit  sufficiente da coprire ogni altro arco (2pt)
- 4.2 (2 pt, recognize 2-colorability) Dire, certificandolo, quale sia il minimo numero di archi da rimuovere per rendere bipartiti i grafi G , G_u e G_q (1pt se corretti tutti i certificati di bicolorazione, 1 pt se ok ogni certificato di ottimalit ).
- 4.3 (3 pt, shortest paths) Rispettando i sensi unici, si riporti la distanza di ciascun nodo dal nodo s (1pt). Dare un albero dei cammini minimi (1pt) e si descriva lo spazio di tali alberi precisando quanti sono (1pt).
- 4.4 (4 pt, max-flow/min-cut) In G , trovare un massimo s, t -flusso (2pt) e un minimo s, t -taglio (2pt).
- 4.5 (2 pt, flow sensitivity) Specifica l'incremento nel massimo valore di flusso ottenibile rimuovendo il vincolo di capacit  del singolo arco e , per ogni arco e per il quale detto incremento   non-nullo (1pt).
- 4.6 (2 pt, certify flow sensitivity) Per uno qualsiasi degli archi identificati al punto precedente esibisci prova che aumentandone la capacit  si possa ottenere il valore di flusso dichiarato (1pt). Certifica anche che l'aumento non   superiore a quanto dichiarato (1pt).
- 4.7 (1 pt, one MST) In G , fornire un albero ricoprente di peso minimo.
- 4.8 (2 pt, MST categorize edges) Etichetta ciascun arco con la lettera A se appartiene a ogni MST, B se a nessuno, C altrimenti. (Se li hai ti conviene usare 3 colori.)
- 4.9 (5 pt, MST certificates) Per ciascuno dei quattro archi incidenti nel nodo m certificare l'etichetta assegnatagli al punto precedente.
- 4.10 (4 pt, count MSTs) Quanti sono gli MST in G ?
- 4.11 (2 pt, max match) Fornire un matching di massima cardinalit  in $G_{p,n}$, il grafo ottenuto da G rimuovendo i nodi p ed n (1pt). Certifica la non esistenza di un matching con un numero maggiore di archi? (1pt)

Quadro delle risposte dell'Esercizio 4

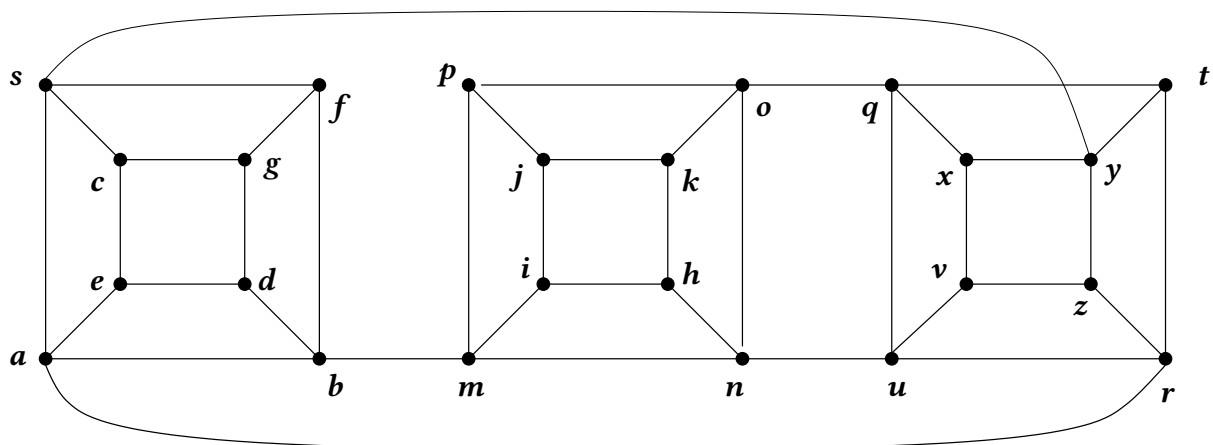


Figura 1: **planarità o meno di G .**

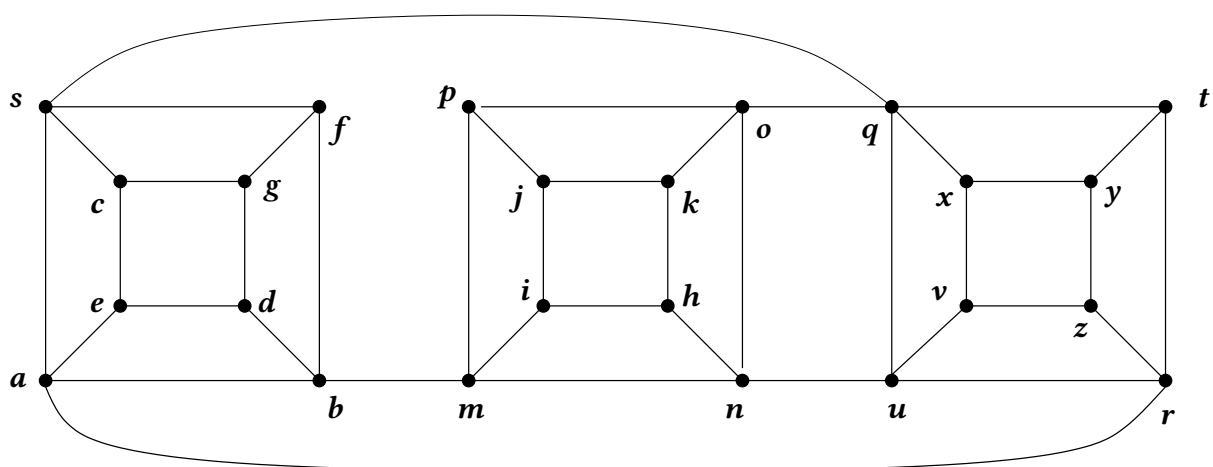


Figura 2: **planarità o meno di G_q .**

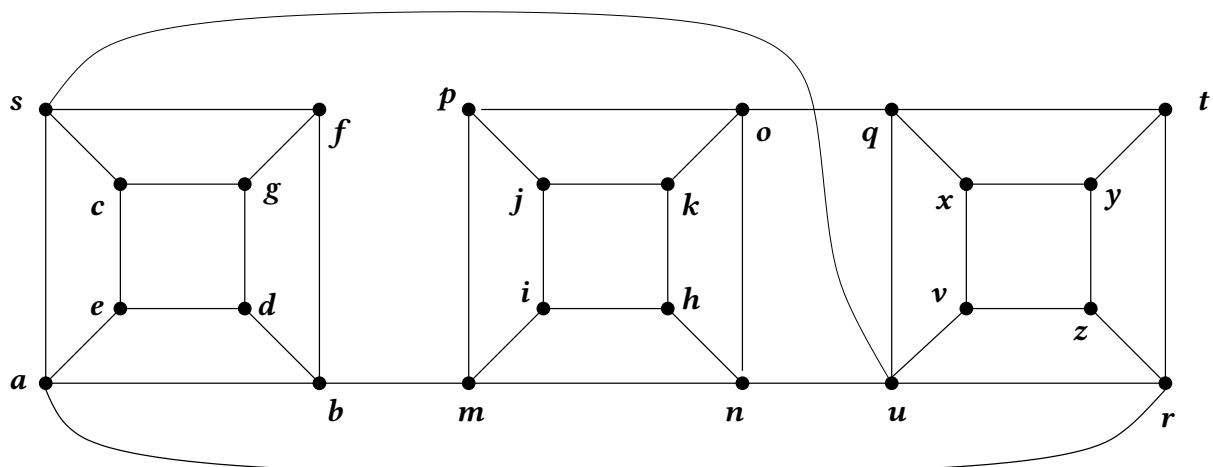
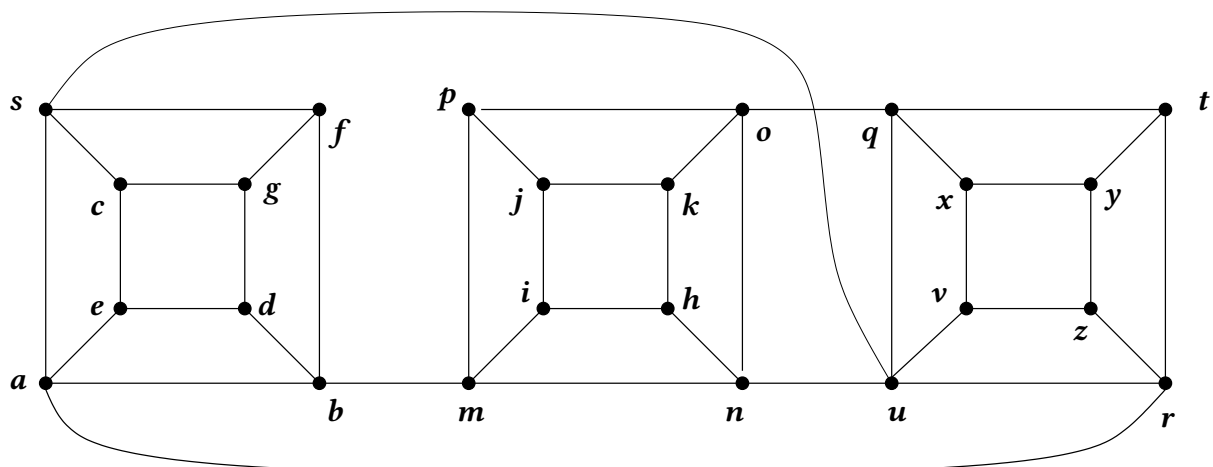
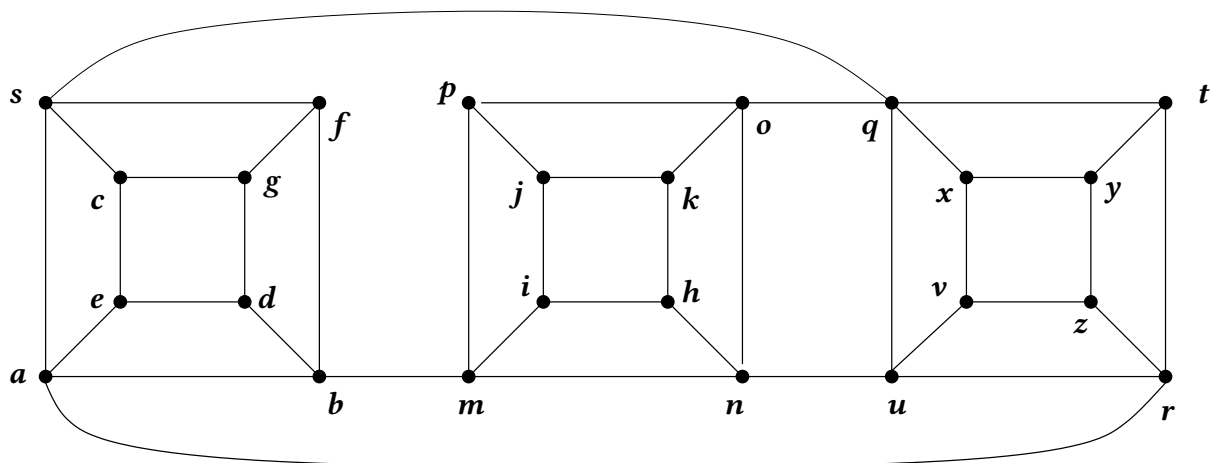
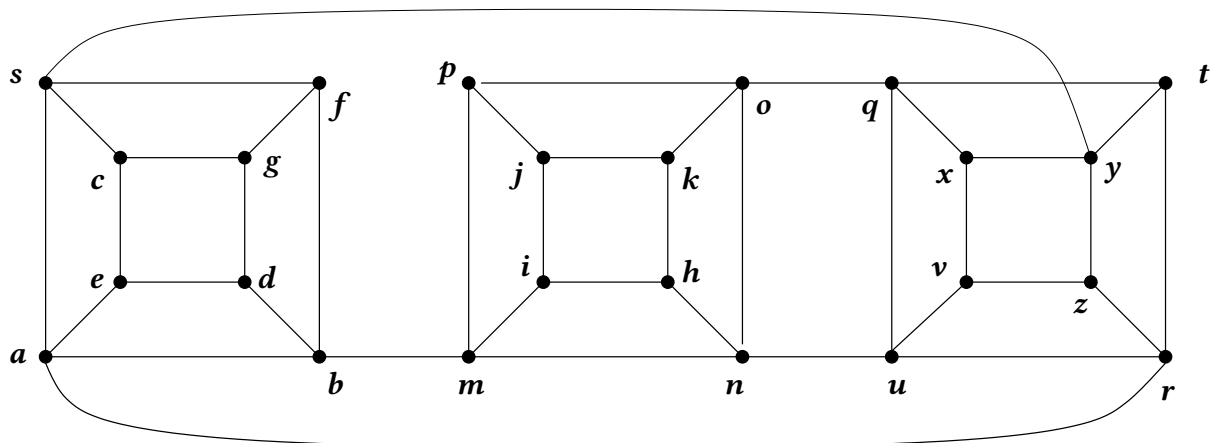
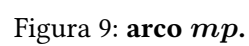
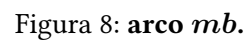
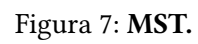


Figura 3: **planarità o meno di G_u .**





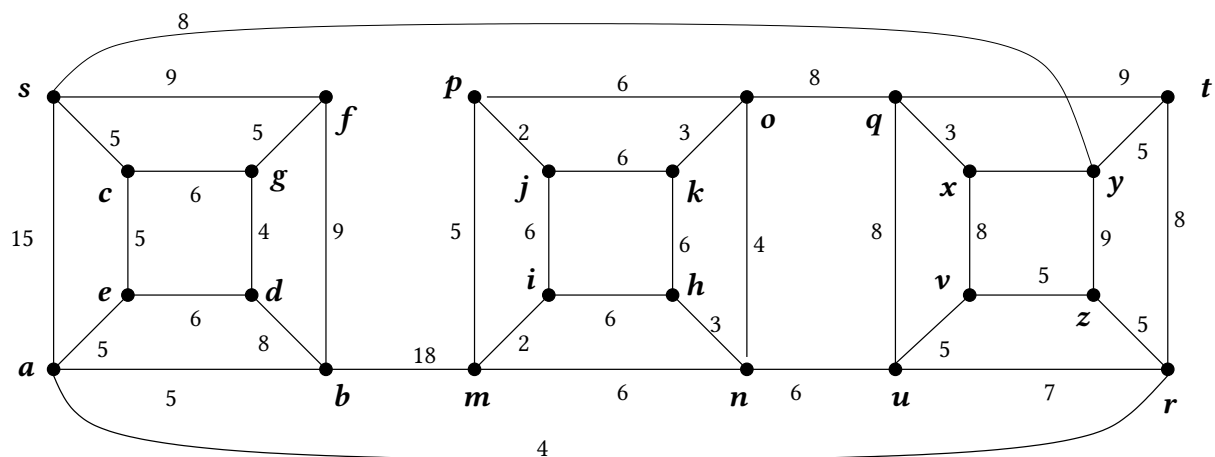


Figura 10: arco mi .

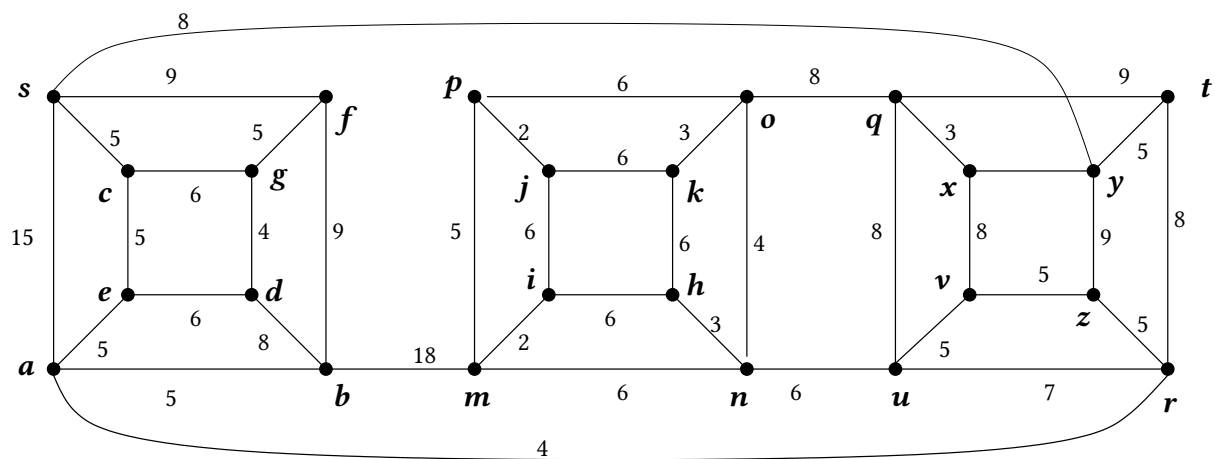


Figura 11: arco mn .

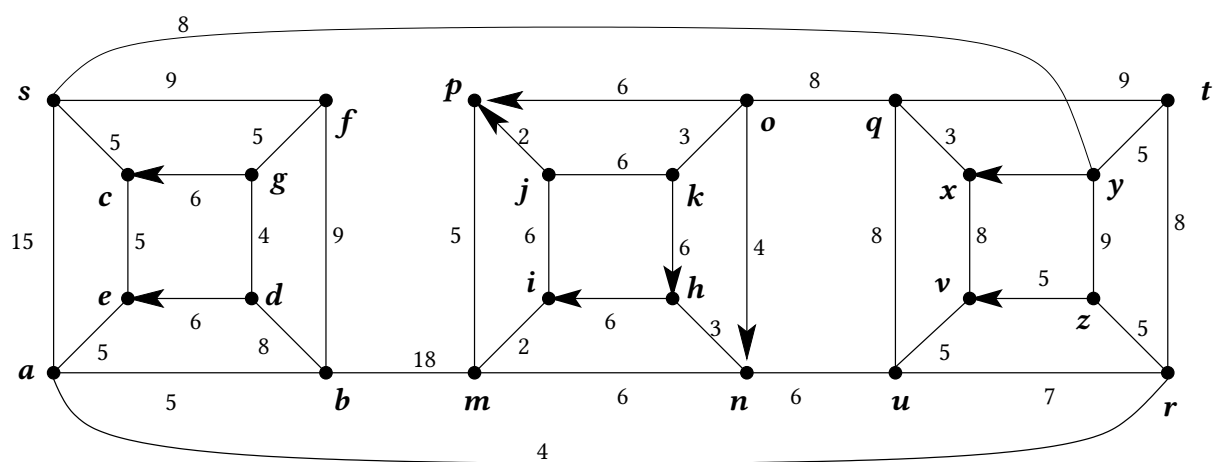


Figura 12: max-flow min-cut.

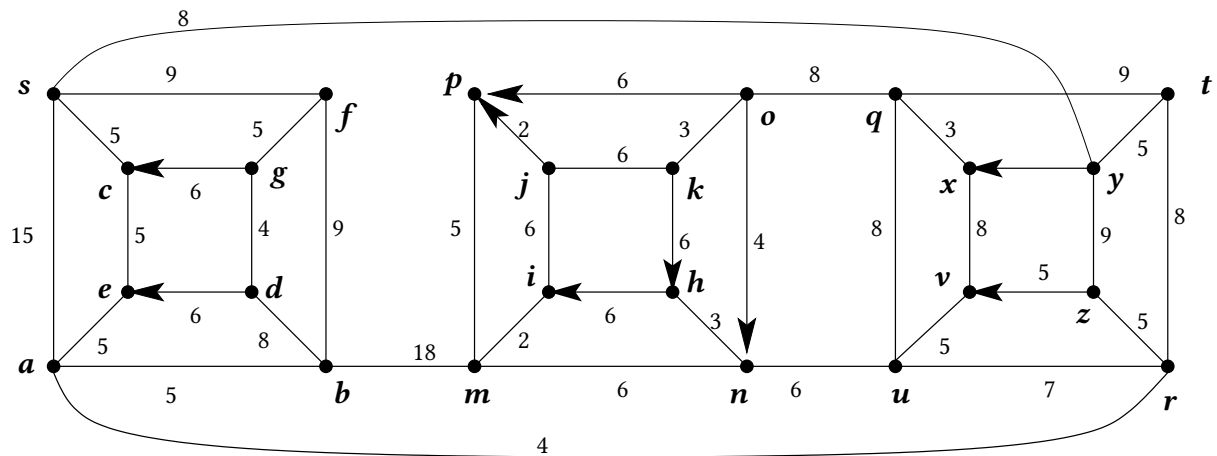


Figura 13: certificati di corretta valutazione della sensitività per l'arco scelto.

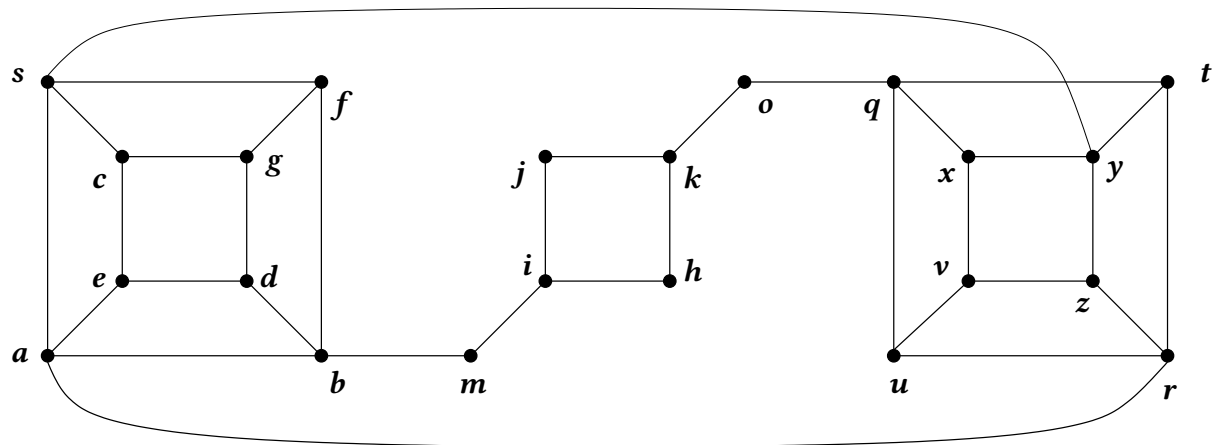


Figura 14: max matching.

LEGGERE CON MOLTA ATTENZIONE:

Procedura da seguire per l'esame -collaborare al controllo

1) Vostro nome, cognome e matricola vanno scritti, prima di incominciare il compito, negli appositi spazi previsti nell'intestazione di questa copertina. Passando tra i banchi verificherò la corrispondenza di queste identità. Ulteriori verifiche alla consegna.

2) Ripiega questa copertina a mo' di teca (intestazione coi dati personali su faccia esterna). In essa inserirai i fogli col tuo lavoro per raccogliarli. Vi conviene (non richiesto) che anche essi riportino Nome/Cognome/Matricola per scongiurare smarrimenti. Conviene consegnare tutto quanto possa contenere ulteriore valore (potete tirare una riga su inutili ripetizioni, risposte sbagliate, parti obsolete).

3) **non consentito:** utilizzare sussidi elettronici, consultare libri o appunti, comunicare con i compagni.

4) Una volta che sono stati distribuiti i compiti non è possibile allontanarsi dall'aula per le prime 2 ore. Quindi: non venite all'esame solo per fare i curiosi con quella di uscirvene quando vi pare (testi e correzione vengono pubblicati a valle dell'esame) oppure portatevi altre cose da fare in quelle ore.

Procedura da seguire per ogni esercizio -assegnazione punti

1) Assicurarsi di fornire i certificati idonei ovunque richiesti.

2) Trascrivere i risultati ottenuti negli appositi riquadri ove previsti.

Comunicazione esiti e registrazione voti -completamento esame

I voti conseguiti restano validi fino ad eventuale consegna ad un qualche appello successivo. La registrazione dell'ultimo voto conseguito va richiesta come da dettagli nella comunicazione degli esiti.