NOME:	COGNOME: .....
MATRICOLA:	FIRMA: .....

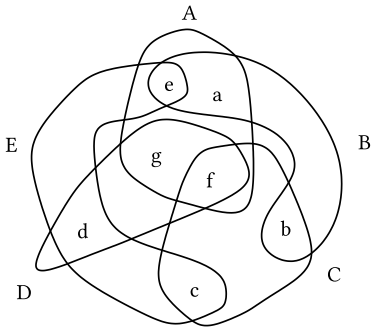
Esame di Ricerca Operativa - 24 giugno 2024

Facoltà di Scienze MM.FF.NN. - Verona

4 esercizi per 79 punti in palio (voto $\geq$ punti -6, $40 \longrightarrow 30$ *e lode*)

Esercizio 1 (con 12 richieste: 3+1+3+2+1+4+2+3+1+5+1+5 = 31 punti [modellazione/riduzioni]):

Un ipergrafo è una coppia $H = (U, \mathcal{E})$ con U un insieme finito di elementi chiamati *nodi* ed $\mathcal{E}$ una famiglia di sottoinsiemi non-vuoti di U chiamati *iperarchi*. In pratica H è una qualsiasi famiglia di insiemi non-vuoti, e viene chiamato anche col nome di *set system*. Un *set cover* di H è un qualsiasi sottoinsieme $\mathcal{E}'$ di $\mathcal{E}$ tale che $\cup_{e \in \mathcal{E}'} e = U$.



esempio. i set covers dell'ipergrafo $\overline{H}$ in figura sono $\{A, B, E\}$, $\{A, C, D\}$, $\{A, C, E\}$, $\{B, D, E\}$ e tutti i loro sovrainsiemi. Non è invece set cover di $\overline{H}$ alcun sottoinsieme di $\{A, B, C\}$ (il nodo d rimarrebbe scoperto), di $\{A, B, D\}$ (manca c), di $\{A, D, E\}$ (manca b), di $\{B, C, D\}$ (manca e), di $\{B, C, E\}$ (manca g), o di $\{C, D, E\}$ (manca a).

MIN SET COVER è il problema di trovare un set cover di minima cardinalità per un generico ipergrafo H dato in input.

Richieste dell'Esercizio 1

1.1 (3 pt, model via hypergraphs) Un caporeparto ha preso atto che ci sono coppie di dipendenti che quando presenti entrambi passano tutto il giorno alla macchinetta del caffè a parlare male di tè o indugiano in altri comportamenti negativi non più tollerabili. Formula in termini di MIN SET COVER il problema di chi licenziare assumendo di voler tenere più personale possibile visto che si è investito così tanto su di loro nei corsi di formazione per insegnargli a lavorare di più a meno paga.

1.2 (1 pt, forge hypergraph model) Ad ogni dipendente hai associato un valore che esprime quanto ti costerebbe licenziarlo. Prova a definire un problema MIN SET COVER PESATO che ti consenta di meglio rappresentare il problema di individuare chi licenziare volendo minimizzare i costi.

1.3 (3 pt, model as ILP) Formula come un problema di Programmazione Lineare Intera (PLI) il problema MIN SET COVER per la specifica istanza $\overline{H}$ in figura.

1.4 (2 pt, generalize ILP model) Estendi la tua formulazione PLI a un generico ipergrafo $H = (U, \mathcal{E})$.

1.5 (1 pt, model as ILP) Offri una formulazione PLI di MIN SET COVER PESATO per istanza generica.

1.6 (4 pt, relax and yin yang) Il rilassamento continuo di un problema di PLI è il problema di Programmazione Lineare (PL) che si ottiene facendo cadere i vincoli di interezza. Per la specifica istanza $\overline{H}$ in figura, scrivi il rilassamento continuo (1pt) della formulazione di PLI che hai dato, e il suo problema duale (2pt). Da quale teorema (1pt) puoi dedurre che se le soluzioni ottime di un problema di PLI restano ottime anche per il rilassamento continuo allora puoi certificarne l'ottimalità esibendo una soluzione ammissibile di pari valore nel duale del problema rilassato?

1.7 (2 pt, general dual) Fornisci il duale del rilassamento continuo per ipergrafo H generico.

1.8 (3 pt, fractional point) Fornire un ipergrafo $H = (U, \mathcal{E})$ per il quale il rilassamento continuo non ha soluzioni ottime che siano intere. Si ottengono almeno 2pt se si fornisce una soluzione frazionaria ottima e si argomenta che ogni soluzione intera costerebbe di più, 3pt se $|U|$ è la più piccola possibile.
1.9 (1 pt, model via graphs) Questa richiesta di un solo punto è un espediente che potrebbe facilitarti nell'intuire una riduzione da Min Node Cover a Max Independent Set, che è l'oggetto della più ambiziosa richiesta successiva. Un *node cover* di un grafo $G = (V, E)$ è un sottoinsieme di V che contenga almeno uno dei due estremi per ciascun arco in E. Min Node Cover è il problema di trovare un node cover di minima cardinalità per un generico grafo G dato in input. Domanda: Come potresti esprimere il tuo dilemma di quale personale licenziare e quale invece mantenere in termini di Min Node Cover?
1.10 (5 pt, NP-hardness proof 1) Min Node Cover è noto essere NP-hard. Sfrutta questo fatto per dimostrare che anche Min Set Cover è NP-hard.
1.11 (1 pt, NP-hardness proof 2) Deduci dal risultato sopra che anche Min Set Cover Pesato è NP-hard.
1.12 (5 pt, NP-hardness proof 3) SAT è noto essere NP-hard. Sfrutta questo fatto per dimostrare che anche Min Set Cover è NP-hard.

Esercizio 2 (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 seconda 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**
66	45	54	85	95	52	63	74	80	51	71	94	56	77	65	90	92	55	57	49	72	91	59	82	92	75	58

Richieste dell'Esercizio 2
2.1 (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 decrescente di S che di S prenda l'elemento in posizione i come suo ultimo elemento.
2.2 (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 decrescente di S che di S prenda l'elemento in posizione i come suo primo elemento.
2.3 (1 pt, opt_sol: libera) Trovare una sottosequenza strettamente decrescente di S di massima lunghezza. Specificare quanto è lunga e fornirla.
2.4 (1 pt, certify opt) Fornire un minimo numero di sottosequenze mai decrescenti tali che ogni elemento della sequenza originale in input ricada in almeno una di esse. Specificare quante sono e fornirle.
2.5 (1 pt, opt_sol: last) Trovare una sottosequenza strettamente decrescente di S di massima lunghezza tra quelle che terminano con l'elemento in posizione 19. Specificare quanto è lunga e fornirla.
2.6 (1 pt, opt_sol: left) Trovare una sottosequenza strettamente decrescente 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.
2.7 (1 pt, opt_sol: prende) Trovare una sottosequenza strettamente decrescente di S di massima lunghezza tra quelle che di S prendono l'elemento in posizione 13. Specificare quanto è lunga e fornirla.
2.8 (1 pt, Z-sequenza) Una sequenza è detta una Z-sequenza, o sequenza strettamente decrescente 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 minore dell'elemento che lo precede. Trovare la più lunga Z-sequenza che sia una sottosequenza della sequenza data. Specificare quanto è lunga e fornirla.
2.9 (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 decrescenti di S di massima lunghezza.
2.10 (1 pt, opt_sol: last) Quante sono le sottosequenze strettamente decrescenti di S di massima lunghezza tra quelle che prendono l'elemento in posizione 19 come loro ultimo elemento?
2.11 (1 pt, opt_sol: first) Quante sono le sottosequenze strettamente decrescenti di S di massima lunghezza tra quelle che prendono l'elemento in posizione 8 come loro primo elemento?
2.12 (2 pt, quante opt sol: con) Quante sono le sottosequenze strettamente decrescenti di S di massima lunghezza tra quelle che di S includono l'elemento in posizione 13?

Esercizio 3 (con 7 richieste: 3+4+4+4+2+5+4 = 26 punti [grafi]):
Ogni richiesta che faremo può essere agevolmente evasa utilizzando la *rappresentazione a liste di adiacenza* del grafo in questione. Per ogni nodo, si fornisce la lista degli archi entranti e quella degli archi uscenti, entrambe terminate da ";" e quando una è vuota resta indicata da un trattino ("-"). Ogni arco ricompreso nelle due liste associate ad un *nodo* è una coppia (*altro nodo*, *costo*). Alcune delle richieste che seguono si riferiscono al grafo non-diretto e/o non-pesato ottenuto ignorando semplicemente l'orientazione o il peso dell'arco.

A: (F,4), (J,2); (U,3), (B,5);
B: (A,5), (S,5); (C,5), (U,4), (G,4);
C: (F,5), (H,5), (B,5); (K,4);
D: (F,2); (U,5), (W,3), (R,3);
E: (O,4), (T,4); (P,4);
F: (K,5); (C,5), (D,2), (I,3), (A,4);
G: (B,4); (H,4), (J,4);
H: (G,4); (C,5), (I,4);
I: (F,3), (H,4); (J,3);
J: (I,3), (G,4); (A,2);
K: (C,4), (S,5); (F,5);
L: (M,1), (N,1); -;
M: (S,2); (L,1);
N: (S,2); (L,1);
O: -; (P,4), (E,4);
P: (O,4), (E,4); (Q,3);
Q: (P,3); (T,3);
R: (D,3), (X,5), (Z,4); -;
S: -; (B,5), (M,2), (N,2), (K,5);
T: (Q,3); (E,4);
U: (D,5), (A,3), (B,4); (V,5), (Z,3);
V: (U,5); (W,5), (Y,3);
W: (D,3), (V,5); (X,4);
X: (W,4); (Y,4), (R,5);
Y: (V,3), (X,4); (Z,5);
Z: (U,3), (Y,5); (R,4);

Di massima, i vari elementi in risposta ad una richiesta potranno essere forniti compilando una diversa riga di una tabella la cui prima riga contiene i nomi dei nodi in ordine alfabetico. Se preferisci tenere separare le tabelle delle risposte per le varie richieste, assicurati che ciascuna abbia tali intestazioni di colonna per non complicare gli atti di verifica.

Ad esempio, la prima richiesta chiede di individuare le componenti connesse del grafo (non-diretto e non-pesato), e sia la BFS che la DFS (puoi scegliere) sono due semplici ed efficaci (lineari) algoritmi cui basta la rappresentazione a liste di adiacenza per produrre ogni elemento di risposta richiesto.

Richiesta di studiare le componenti connesse. In un grafo non-diretto $G = (V, E)$ due nodi si dicono *collegati* se e solo se tra di essi vi è un cammino. Essere collegati è una relazione di equivalenza le cui classi sono chiamate le *componenti connesse* di G. Di fatto, un $V' \subseteq V$ e il sottografo indotto $G[V']$ sono entrambi chiamati componenti connesse di G se V' è massimale con la proprietà che ogni suoi due nodi siano collegati. Non ti serve disegnare G per esprimere efficacemente la partizione di V in componenti connesse: in una tabella delle risposte che ha per etichette di colonna i nomi dei nodi presi in ordine alfabetico, e scelto come rappresentante di ciascuna classe il suo primo nodo in ordine alfabetico, la partizione in componenti connesse può essere espressa con una sola riga (1pt) della tabella che per ogni nodo v in V riporta il nome del rappresentante della classe di v. Analizziamo l'effi-

cacia di tale codifica: per verificare che due nodi cui hai attribuito rappresentanti diversi appartengano effettivamente a classi diverse basterà controllare che i due estremi di ogni arco dichiarino lo stesso rappresentante. Pertanto, su questo versante, non serve aggiungere un certificato. Ma come verificare che tutti i nodi cui sia stato assegnato uno stesso rappresentante x siano effettivamente collegati ad x? Sia io (Arturo) che tù (Merlino) vogliamo assicurarci che la vera partizione di V in componenti connesse non sia in realtà una sotto-partizione di quella che tu consegni. Collaborare è una risorsa. L'accordo è che per ogni componente connessa $V_x \subseteq V$ Merlino fornisca inoltre un albero ricoprente del sottografo indotto $G[V_x]$. Un modo pratico per entrambi di passarsi questo albero è di orientarlo in modo che abbia radice in x, in questo modo ogni nodo $v \neq x$ avrà un unico padre. Pertanto, in una seconda riga (1pt) della tabella, ogni nodo riporterà l'identità del proprio padre (le radici, ossia i rappresentanti delle rispettive classi di equivalenza, riporteranno semplicemente sè stesse). Come ulteriore garbo ad Arturo (1pt), potresti fornire una terza informazione certificante (dell'aciclicità dei puntatori al padre) specificando, per ogni nodo, la sua distanza generazionale dalla rispettiva radice (ossia la lunghezza del cammino che si otterrebbe risalendo alla radice seguendo i padri).

Richieste dell'Esercizio 3

3.1 (3 pt, componenti connesse) Identificazione delle componenti connesse (1 punto per ogni elemento o co-elemento certificante, ossia per ogni riga della tabella delle risposte come precisata sopra).

3.2 (4 pt, 2-colorability) Garantiamo che ogni componente connessa possa essere resa bipartita con la rimozione di al più 1 arco (fù mio errore, in realtà erano 2). Per le componenti connesse non-bipartite fornire un ciclo dispari (1pt) e indicazione dell'arco da rimuovere (1pt). Fornire, in una riga della tabella delle risposte, una 2-colorazione $c : V \longrightarrow \{0, 1\}$ valida a valle della rimozione degli archi indicati (2pt se colori tutti i nodi, 1 se colori almeno tutti quelli delle componenti già in partenza bipartite).

3.3 (4 pt, cammini minimi) In una riga della tabella si riporti la distanza (1pt) di ogni nodo da S. Si fornisca un albero dei cammini minimi dal nodo S ai nodi raggiungibili da S (per ogni nodo il padre, (1pt)). Fuori dalla tabella, si riporti la lista delle scelte alternative per quei nodi che potrebbero optare per un altro padre (1pt), e dire quanti sono i diversi alberi dei cammini minimi (1pt).

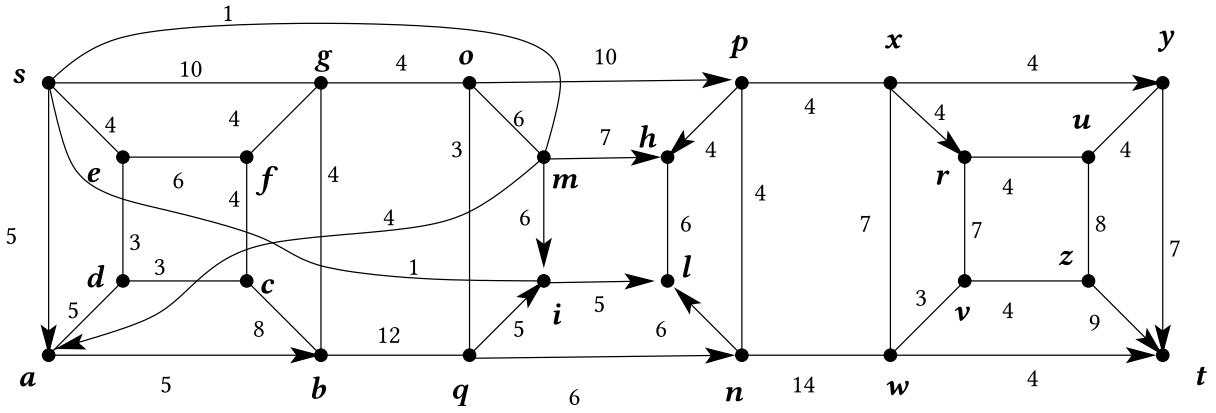
3.4 (4 pt, cammini minimi diretti) La stessa consegna di cui al punto precedente, e con lo stesso schema attributivo dei punti, ma ora con riferimento al grafo diretto. Per ogni lista che hai ricevuto in input, prima del ";" hai gli archi entranti nel nodo e dopo il ";" quelli uscenti.

3.5 (2 pt, MST: algoritmo di Prim) Sia $\overline{C}$ la componente connessa di G contenente il nodo S. Si fornisca un albero ricoprente di peso minimo per $\overline{C}$, orientato con radice in S (1pt per la riga dei padri, 1pt per la distanza generazionale da S).

3.6 (5 pt, MST: cicli e tagli, 5=1+1+1+2) Per ciascuno degli archi IH, IJ e ZR, dire (1pt se ogni arco è classificato correttamente) se esso appartenga a tutti, oppure a nessuno, oppure a qualcuno ma non tutti gli alberi ricoprenti di peso minimo di $\overline{C}$, fornendo i cicli e/o tagli certificati (1pt per ogni certificato non-ridondante). Organizzarsi per fornire in modo chiaro ogni elemento di risposta (non si prestano ad essere espressi con una riga di tabella delle risposte come per altre richieste).

3.7 (4 pt, DAG recognition) Con riferimento al grafo diretto, garantiamo che ogni componente connessa V_x possa essere resa aciclica rimuovendo al più un arco. Per ogni componente connessa che contenga cicli diretti se ne fornisca uno (1pt) e si dia indicazione dell'arco da rimuovere (1pt) per renderla aciclica. In una riga della tabella si fornisca un ordinamento topologico del grafo ottenuto con la rimozione degli archi indicati, ossia una numerazione dei nodi tale che per ogni arco il numero assegnato al nodo testa superi il numero assegnato al nodo coda. (2pt se lo fai per tutte le componenti, 1 se gestisci solo quelle che erano dei DAG già in partenza).

Esercizio 4 (con 3 richieste: 3+2+3 = 8 punti [grafi]):



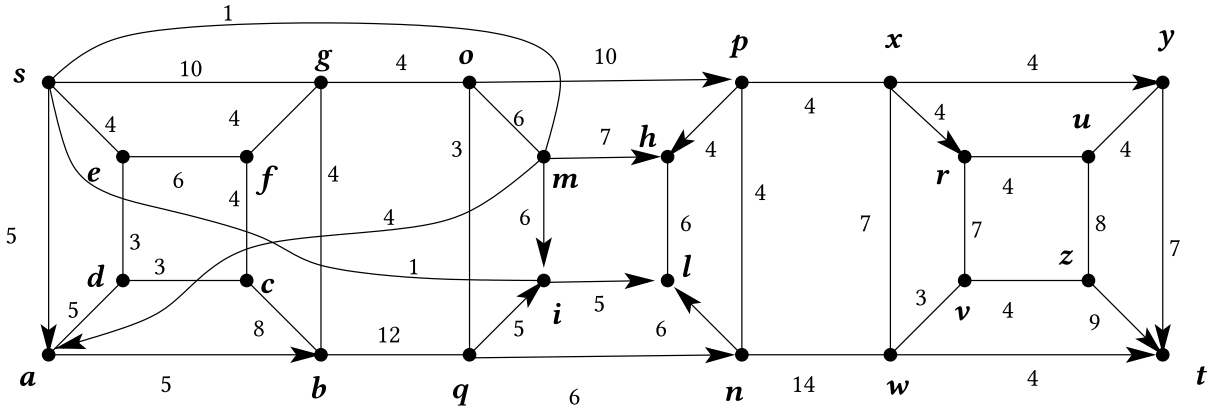
Richieste dell'Esercizio 4

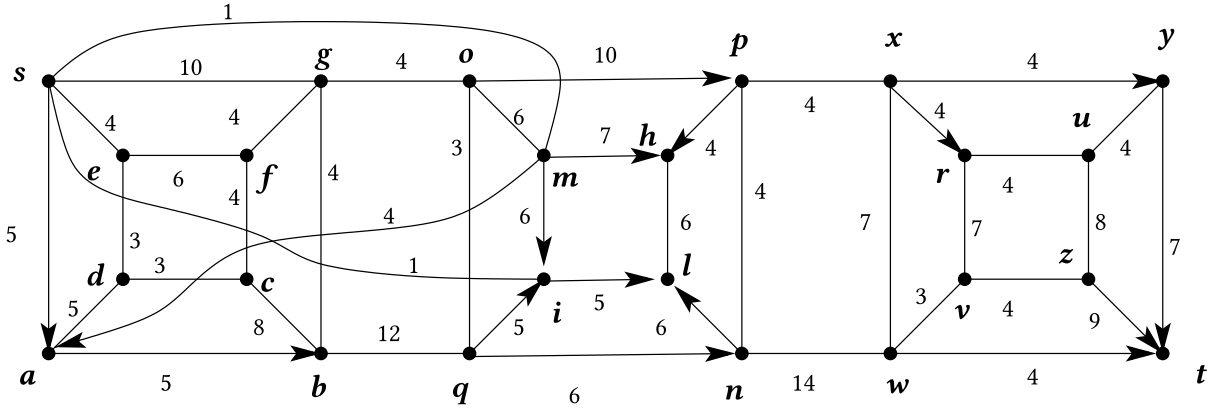
4.1 (3 pt, recognize planarity) Dire, certificandolo, se siano planari o meno il grafo G e il grafo G' ottenuto da G sostituendo l'arco si con un arco so. (2 punti per il certificato di non-planarità, 1 per quello di planarità)

4.2 (2 pt, max flow) In G, trovare un massimo flusso dal nodo s al nodo t.

4.3 (3 pt, min cut) In G, trovare un s,t-taglio minimo.

ulteriore supporto per risposte.





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 raccoglierli. 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: (1) andate al bagno prima della distribuzione dei compiti, (2) portatevi snacks e magliocino (specie nei laboratori, specie in estate, stando fermi a lungo si patisce il freddo), e (3) 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 và richiesta come da dettagli nella comunicazione degli esiti.