

NOME:

COGNOME:

MATRICOLA:

FIRMA:

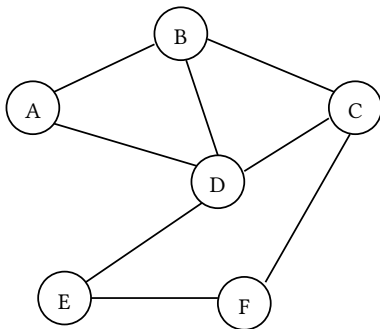
Esame di Ricerca Operativa - 14 settembre 2023

Facoltà di Scienze MM.FF.NN. - Verona

4 esercizi per 75 punti in palio (voto \geq punti $-6, 40 \rightarrow 30$ e lode)

Esercizio 1 (con 11 richieste: $2+3+2+1+1+5+3+3+1+5+1 = 27$ punti [modellazione/riduzioni]):

Un sottoinsieme dei nodi $S \subseteq V$ è un *node cover* di un grafo $G = (V, E)$ se contiene almeno uno dei due estremi di ciascun arco $uv \in E$. Quando invece, per nessun arco $uv \in E$, l'insieme S contiene entrambi gli estremi, allora S è un *independent set* di G .



esempio. nel grafo G in figura, $\{A, C, D, F\}$ e $\{B, D, F\}$ sono due node covers mentre $\{C, D, E, F\}$ non lo è perchè non contiene nè l'estremo A nè l'estremo B dell'arco AB . Invece $\{B, F\}$, $\{D, F\}$ e $\{A, E\}$ sono tre independent set mentre $\{A, C, F\}$ non lo è per via dell'arco CF .

MIN NODE COVER è il problema di trovare un node cover di minima cardinalità per un generico grafo G dato in input.

MAX INDEPENDENT SET è il problema di trovare un independent set di massima cardinalità per un grafo G dato in input.

Richieste dell'Esercizio 1

1.1 (2 pt, model via graphs) 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 NODE 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 (3 pt, model as ILP) Formula come un problema di Programmazione Lineare Intera (PLI) il problema MIN NODE COVER per la specifica istanza G in figura.

1.3 (2 pt, generalize model) Estendi la tua formulazione PLI a un generico grafo $G = (V, E)$.

1.4 (1 pt, forge graph model) Ad ogni dipendente hai associato un valore che esprime quanto ti costerebbe licenziarlo (dipende dall'età). Prova a definire un problema MIN NODE COVER PESATO che ti consenta di meglio rappresentare il problema di individuare chi licenziare volendo minimizzare i costi.

1.5 (1 pt, model as ILP) Offri una formulazione PLI di MIN NODE COVER PESATO per grafo generico.

1.6 (5 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 G in figura, scrivi il rilassamento continuo 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 esibire una soluzione ammissibile del duale con lo stesso valore, che pertanto certifichi l'ottimalità di ciascuna soluzione ottima del problema intero? Fornisci una coppia di soluzioni intere, una per il primale ed una per il duale (2pt), che certifichino vicendevolmente la propria ottimalità.

- 1.7 (3 pt, general dual) Fornisci il duale del rilassamento continuo per un grafo generico $G = (V, E)$.
- 1.8 (3 pt, fractional opts) Nel caso del sottografo ottenuto dal grafo in figura per rimozione dei nodi C, E, F il rilassamento continuo e il suo duale non hanno soluzioni ottime intere. Argomenta perché nessuno dei due può avere soluzioni ottime intere (1pt) dopo aver fornito una soluzione ottima per entrambi (2pt). (Non serve che usi il semplice, puoi intuirle direttamente dato che poi si autocertificano vicendevolmente.)
- 1.9 (1 pt, model via graphs) Questa richiesta di un solo punto è un espediente che potrebbe facilitarti nell'intuire una riduzione da MAX INDEPENDENT SET a MIN NODE COVER, che è l'oggetto della più ambiziosa richiesta successiva. Domanda: Come potresti esprimere il tuo dilemma di quale personale licenziare e quale invece mantenere in termini di MAX INDEPENDENT SET? Sono poi così diversi questi due importanti problemi su grafo?
- 1.10 (5 pt, NP-hardness proof) MAX INDEPENDENT SET è noto essere NP-hard. Sfrutta questo fatto per dimostrare che anche MIN NODE COVER è NP-hard. Osserva che di fatto i problemi MAX INDEPENDENT SET e MIN NODE COVER sono in tutto equivalenti: non puoi risolvere l'uno senza risolvere l'altro.
- 1.11 (1 pt, NP-hardness proof) Deduci dal risultato sopra che anche MIN NODE COVER PESATO 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	15	60	52	63	74	80	51	71	97	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: (T,5); (H,5), (I,2), (D,3), (F,4);	J: (X,4), (K,4); (Y,4);	S: -; (G,5), (V,2), (W,2), (T,5);
B: (G,4); (C,4), (E,4);	K: (Z,3); (J,4);	T: (H,4), (S,5); (A,5);
C: (B,4); (H,5), (D,4);	L: (I,5), (F,3), (G,4); (M,5), (Q,3);	U: (V,1), (W,1); -;
D: (A,3), (C,4); (E,3);	M: (L,5); (N,5), (P,3);	V: (S,2); (U,1);
E: (D,3), (B,4); (F,2);	N: (I,3), (M,5); (O,4);	W: (S,2); (U,1);
F: (A,4), (E,2); (L,3), (G,5);	O: (N,4); (P,4), (R,5);	X: -; (Y,4), (J,4);
G: (F,5), (S,5); (H,5), (L,4), (B,4);	P: (M,3), (O,4); (Q,5);	Y: (X,4), (J,4); (Z,3);
H: (A,5), (C,5), (G,5); (T,4);	Q: (L,3), (P,5); (R,4);	Z: (Y,3); (K,3);
I: (A,2); (L,5), (N,3), (R,3);	R: (I,3), (O,5), (Q,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'efficacia 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 tu (Merlino) vogliamo assicurarci che la vera partizione di V in componenti connesse non sia in realtà una sotto-partizione di quella che tu consegna. 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 se 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 un arco. 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 $c : V \rightarrow \{0, 1\}$ 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 DC , DE e QR , 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).

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 od con un arco oc . (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.

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: (1) andate al bagno prima della distribuzione dei compiti, (2) portatevi snacks e maglione (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 va richiesta come da dettagli nella comunicazione degli esiti.