

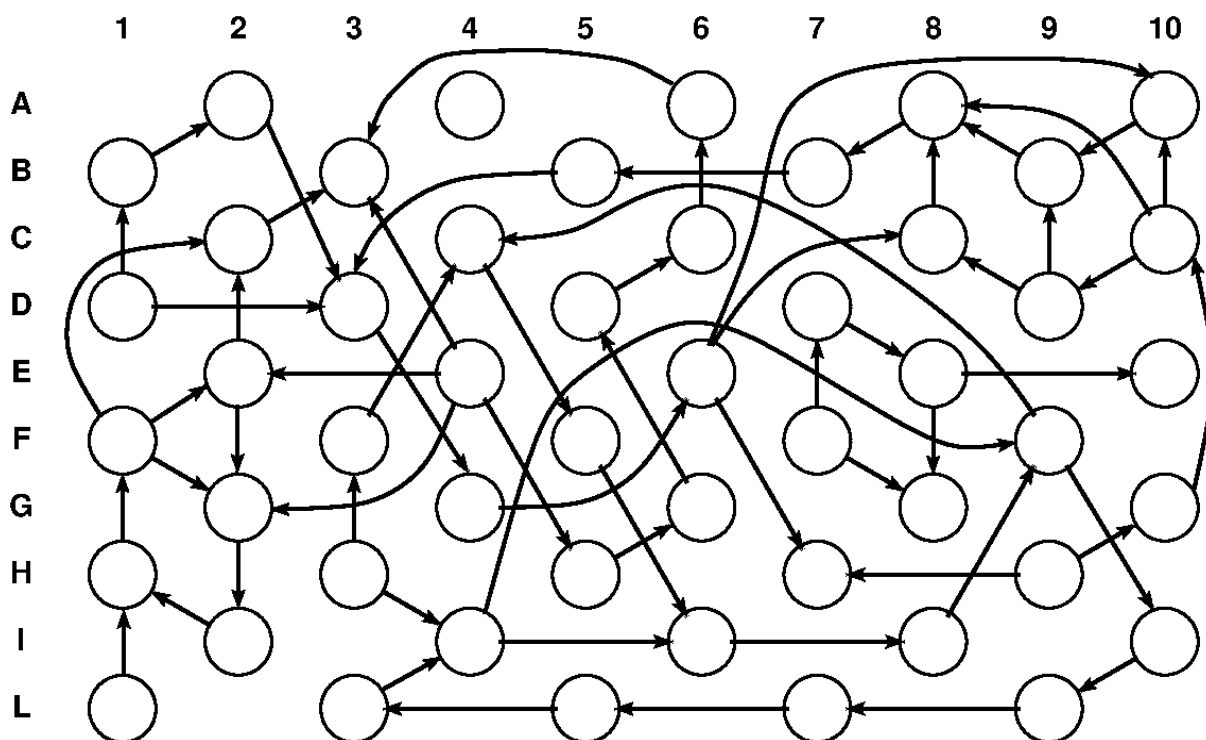
# Esame di Ricerca Operativa - 25 settembre 2025

Facoltà di Scienze MM.FF.NN. - Verona

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

## - CORREZIONE -

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 nisba

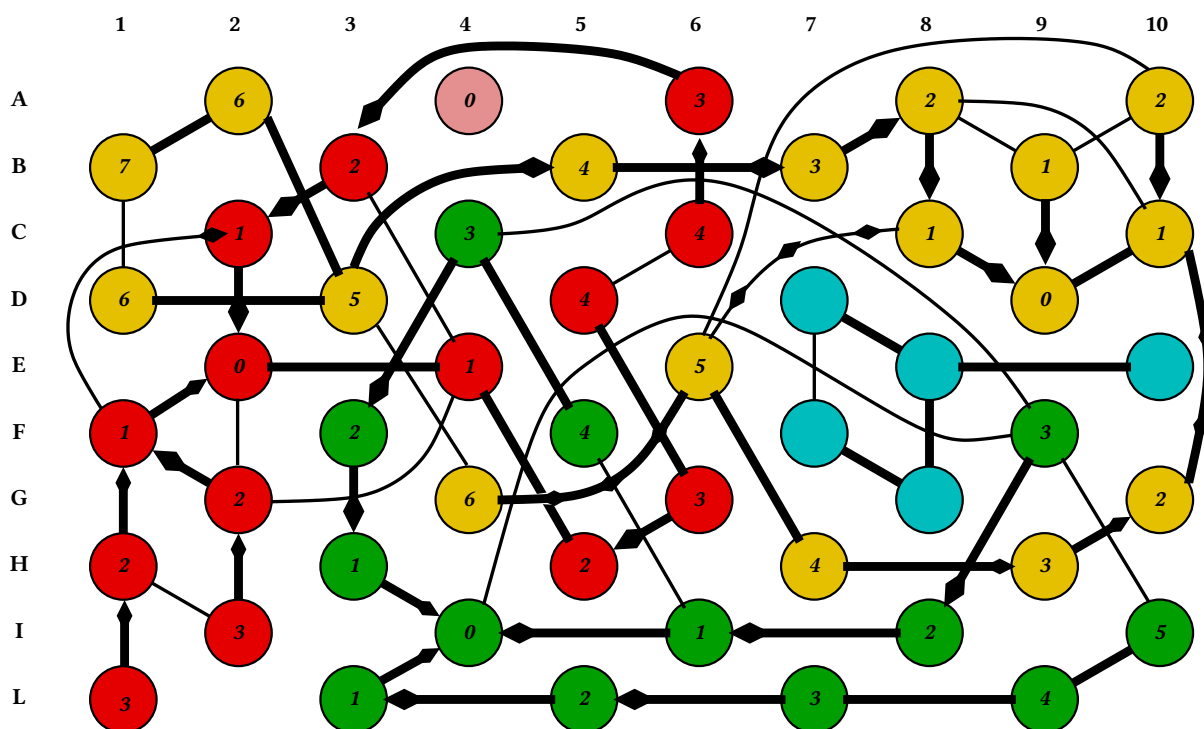
1.6 (1 pt, Eulero) Per ogni componente di più nodi, fornire un ciclo Euleriano se presente, altrimenti un cammino Euleriano se presente, altrimenti spiega perchè nisba

**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)

### Svolgimento esercizio 1 .

**Richiesta 1 (1 pt) (goal: componenti\_connesse).**

La colorazione fornita in figura evidenzia le componenti connesse del grafo. King Arthur può verificare che per ciascun arco i due estremi hanno lo stesso colore e concluderne che, se due nodi hanno colore diverso appartengono allora a componenti diverse.



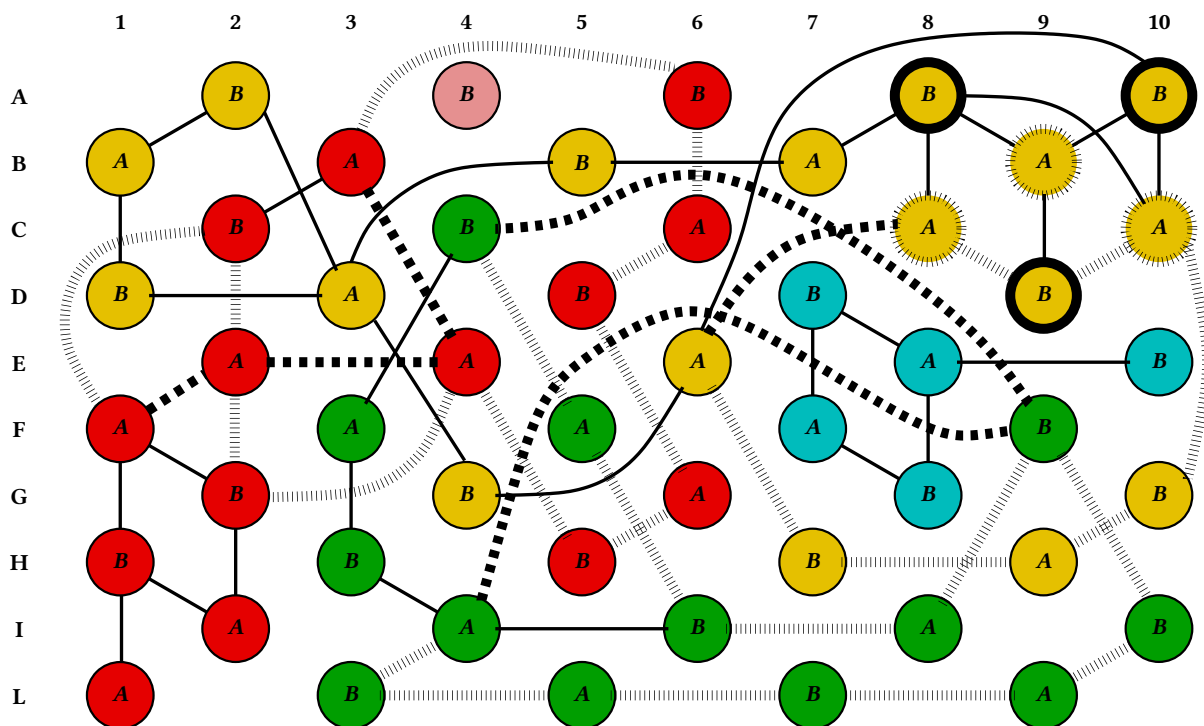
Per certificare l'implicazione inversa (nodi dello stesso colore appartengono ad una stessa componente) abbiamo inoltre fornito uno spanning tree per ogni componente. Per facilitare il lavoro di un King Arthur e consentirgli una verifica locale, negli alberi più complessi abbiamo eletto un rappresentante di classe da ciascuna componente e, fissatolo come radice, abbiamo orientato ogni arco da figlio a padre e riportata inoltre la distanza di ciascun nodo dalla rispettiva radice. Una equipe di King Arthurs che lavorano in parallelo, ciascuno situato su un diverso nodo, potrebbe condurre in  $O(1)$  la seguente verifica: «se la distanza  $d$  riportata dal nodo è zero allora nessun arco esce da detto nodo, altrimenti ne esce precisamente un arco diretto verso un nodo dove è riportata la distanza  $d - 1$ .»

**Richiesta 2 (1 pt) (goal: distingui nodi e archi speciali).**

nodì isolati	foglie	cutnodes	bridges
A1	F10	G3,F8	F8-F10

### Richiesta 3 (3 pt) (goal: make bipartite).

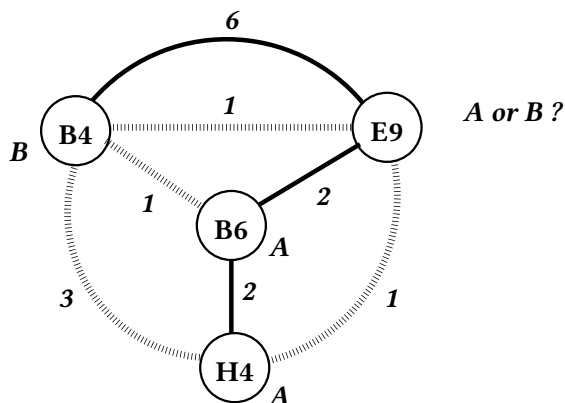
Il problema di produrre una 2-colorazione in cui il minor numero possibile di archi abbiano i due estremi dello stesso colore si decompone naturalmente sulle componenti connesse ed il numero delle soluzioni ottime per il grafo nel suo complesso sarà il prodotto dei numeri di soluzioni ottime per le singole componenti. Una 2-colorazione ottima è offerta in figura, dove ciascun nodo è etichettato A oppure B.



Le componenti rosa e blu sono già bipartite di loro, e la loro bipartizione è unica (a meno di scambiare A e B), ovvero l'unica soluzione ottima consiste nel non rimuovere alcun arco.

Nella componente verde dobbiamo rimuovere almeno due archi come dimostrato dai cicli dispari H4-E5-B6-B8-E9 e B4-A3-A5-A7-A9-B10-E9 evidenziati in figura: il lower-bound disegua poichè tali cicli sono edge-disjoint (ossia non hanno archi in comune) e da ciascuno di essi dobbiamo rimuovere almeno un arco; l'upper bound disegua dalla soluzione ammissibile che prevede di rimuovere i soli archi B4-E9 e E9-H4 rappresentata in figura.

Per analizzare il numero di soluzioni ottime per la componente verde ci ri-rappresentiamo il problema in forma più essenziale qui sulla destra, con un grafo i cui archi stanno per cammini nel grafo originale. Per ogni arco è riportata la lunghezza del cammino che esso sta a rappresentare (se questo valore è un numero dispari, allora l'arco è tratteggiato per rendere più immediatamente percepibile la situazione)



Se, senza perdita di generalità, assumiamo di colorare di A il nodo H4 allora in nodo B4 dovrà essere colorato di B, dato che tra questi due nodi sono presenti  $3 > 2$  cammini dispari edge-disjoint (uno diretto, uno che passa per B6 ed un altro che passa per E9) mentre in nodo B6 dovrà essere colorato A dato che

tra H4 e B6 sono presenti 3 > 2 cammini pari edge-disjoint (uno diretto, uno che passa per B4 ed un altro che passa per E9). Nelle soluzioni ottime dove E9 è colorato A bisognerà rimuovere un qualsiasi arco del cammino H4-E9 (lunghezza 1) e un qualsiasi arco del cammino B4-E9 (quello di lunghezza 6) per un totale di  $1 \times 6 = 6$  possibilità. Le soluzioni ottime dove E9 è colorato B bisognerà rimuovere un qualsiasi arco del cammino B5-E9 (lunghezza 2) e un qualsiasi arco del cammino B4-E9 (quello di lunghezza 1) per un totale di  $1 \times 2 = 2$  possibilità. Sono quindi  $6 + 2 = 8$  le possibilità per la componente verde.

Nella componente gialla dobbiamo rimuovere almeno un arco come dimostrato dal ciclo dispari F6-C7-C9-D10-H10-G9-H8 evidenziato in figura, e la nostra scelta in figura è ricaduta sull'arco H8-F6. Sarebbe stato ottimo rimuovere uno qualsiasi dei seguenti archi: H8-F6, D10-H10, C9-D10, C7-C9, F6-C7; ossia 5 scelte possibili per la componente gialla.

Nella componente rossa dobbiamo rimuovere almeno due archi come dimostrato dai cicli dispari F2-H2-E1 e F4-F2-D2 evidenziati in figura; la nostra scelta in figura è ricaduta sul rimuovere gli archi E1-F2 e D2-F4. L'unica altra opzione ottima era rimuovere gli archi H2-E1 e D2-F2; ossia 2 scelte possibili per la componente rossa.

In conclusione, il numero totale di soluzioni ottime è  $8 \times 5 \times 2$

Numero di Soluzioni Ottime:

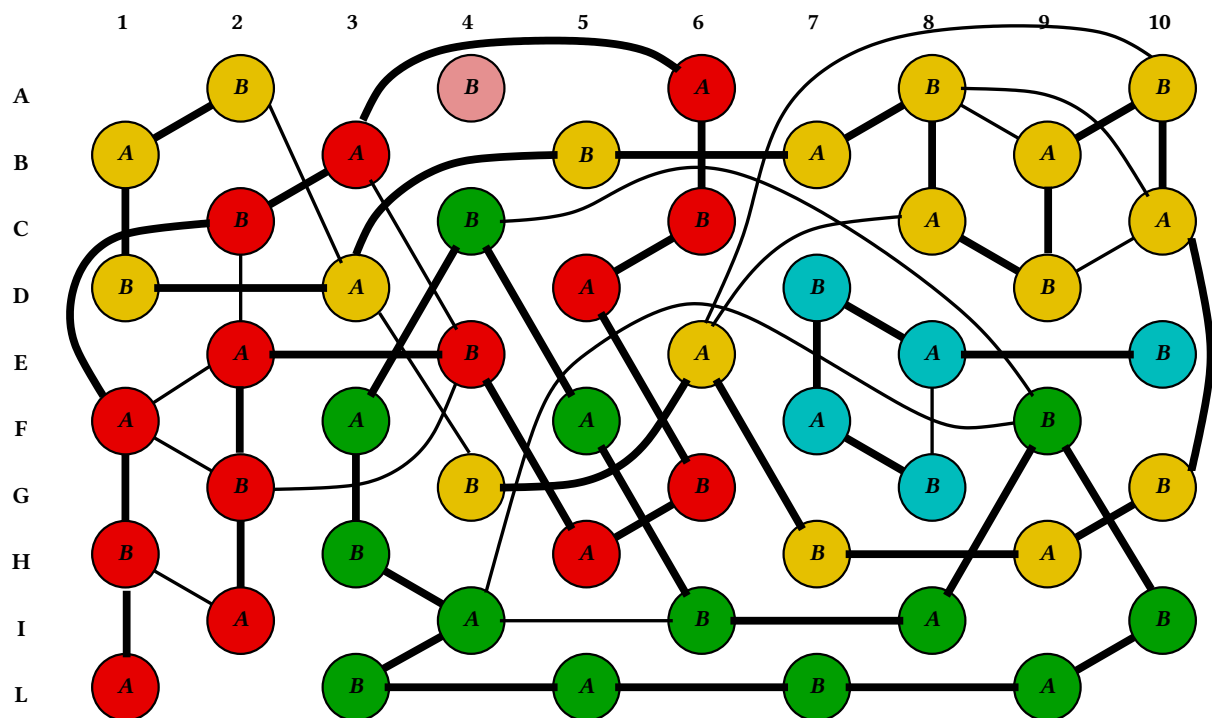
80
----

#### **Richiesta 4 (1 pt) (goal: planarità).**

per argomentare che il grafo non è planare basta argomentare che una qualsiasi delle sue componenti non lo è. Per argomentare che la componente gialla non lo è (invece è facile vedere che ogni altra lo è, solo della verde la figura fornita col testo non esibisce già un planar embedding ma basta ridisegnare l'arco B4-E9 immergendolo nella faccia esterna, ossia facendolo passare sopra tutti gli altri nodi) abbiamo esibito una  $K_{3,3}$ -subdivision nella figura dove i nodi sono stati colorati. Infatti nella stessa figura i sei nodi in basso a destra sono stati evidenziati da un bordo spesso che differenzia inoltre i tre nodi etichettati A (bordo spesso ma tratteggiato) e i tre nodi etichettati B (bordo spesso in linea continua). Questa distinzione corrisponde alla bipartizione del  $K_{3,3}$  in 3 casette e 3 porcellini (acqua, luce e gas). Dobbiamo anche fornire, per ogni casetta, i tre percorsi verso acqua (L8), luce (L10) e gas (G9), avendo cura di non incrociare mai due percorsi su loro nodi interni. La casetta H10 è direttamente collegata (da arco singolo, nessun nodo intermedio) a ciascuna delle tre risorse. Lo stesso dicasi per la casetta I9. La casetta H8 è direttamente collegata ai porcelli L8 e G9 e non è difficile (in P, puoi usare la BFS) individuare un cammino da H8 a L10 nel grafo ottenuto dopo la rimozione degli altri 4 nodi importanti (L8,G9,I9,H10), ad esempio: H8-F6-L10.

#### **Richiesta 5 (1 pt) (goal: Hamilton).**

la seguente figura esibisce un ciclo Hamiltoniano in ciascuna componente che ne possieda almeno uno, altrimenti esibisce un cammino Hamiltoniano chiudendo così la partita.

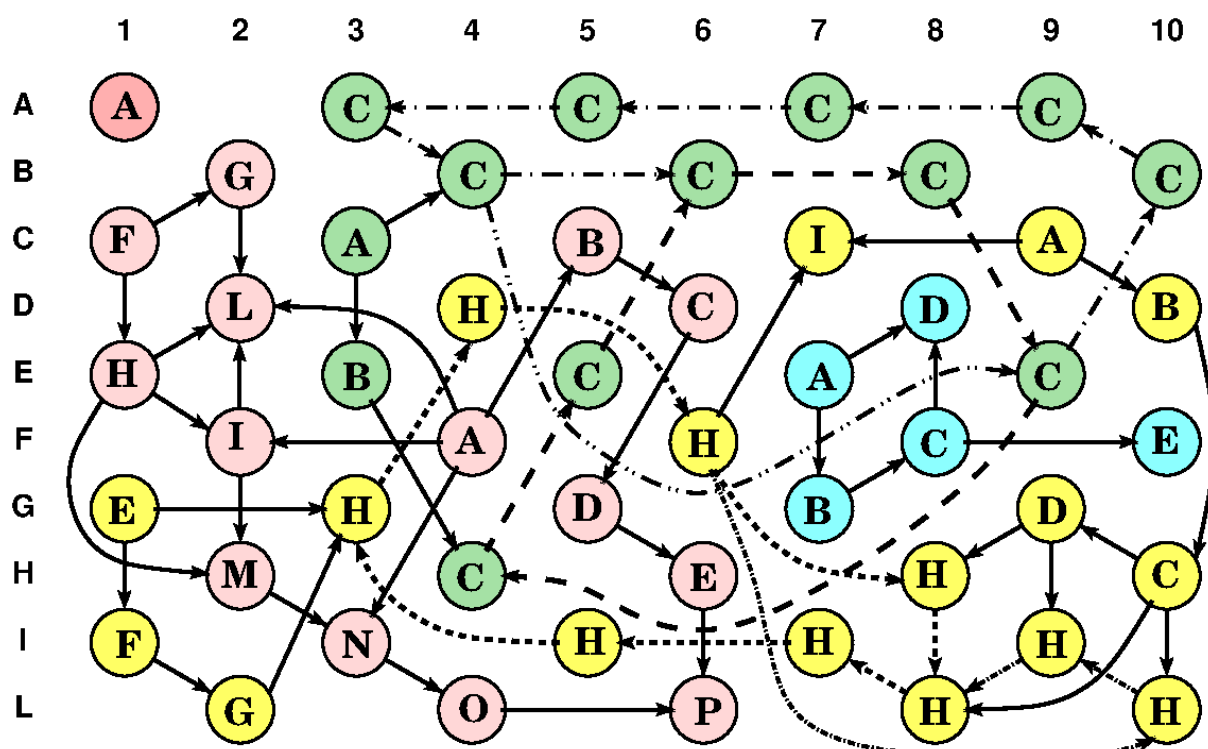


Infatti, la componente blu non può avere cicli Hamiltoniani dato che ha una foglia (F10). La componente gialla non può avere un ciclo Hamiltoniano dato che ha un cutnode (G3).

**Richiesta 6 (1 pt) (goal: Eulero).**

H8, G9 e I9 sono 3 nodi di grado dispari (3) della componente gialla, che pertanto non ha nè cicli nè cammini Euleriani. B6 ed H4 sono gli unici due nodi di grado dispari della componente verde, che pertanto non ha cicli ma ha cammini Euleriani: percorri prima il cammino costituito dai 3 archi che non sono contenuti nel ciclo Hamiltoniano in figura, poi percorri tale ciclo. In modo analogo si ottiene un ciclo Euleriano della componente rossa. La componente blu non può avere cicli Euleriani dato che ha una foglia (F10), ma un cammino Euleriano è ottenuto percorrendo prima l'arco F8-D8 e poi il cammino Hamiltoniano esibito in figura.

**Richiesta 7 (5 pt) (goal: strong-connectivity).**



Le componenti azzurra e rossa sono dei DAG come evidenziato dall'ordinamento topologico  $A, B, C, \dots$  in cui i nodi possono essere eseguiti rispettando le precedenze dettate dagli archi.

Per la componente gialla e verde la figura evidenzia la partizione dei nodi nelle componenti fortemente connesse (due nodi sono labellati con la stessa lettera quando appartengano alla stessa componente, ossia quando ciascuno possa essere raggiunto dall'altro). Il valore della lettera certifica che qualora i nodi di una stessa componente fossero collassati in un unico nodo ne risulterebbe un DAG, ossia che le classi della partizione fornita non vadano ulteriormente fuse tra loro. Che la partizione corretta non sia invece più fine, ossia che ogni due nodi di una stessa classe fornita siano effettivamente mutualmente raggiungibili, è certificato dalla ear-decomposition che abbiamo dato per ogni classe non banale (ossia con almeno due nodi): si costruisca il sottografo indotto dai nodi della classe partendo dal ciclo in linea tratteggiata, poi si aggiungano le orecchie (cammini che partano e terminino in nodi già raccolti nella classe in formazione, ossia già dimostrati mutualmente raggiungibili a questo punto della costruzione; i nodi interni di questi cammini vengono ad aggiungersi alla classe in formazione)

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

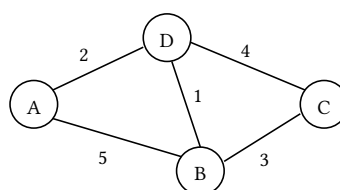
Il modello di PL qui a destra (in alto) computa un albero ricoprente di peso minimo (MST, minimum spanning tree) in un generico grafo  $G = (V, E)$  con costi non-neativi sugli archi  $c : E \rightarrow \mathbb{R}_+$ . Il modello coinvolge un vettore  $x$  di variabili reali, indicizzato dal set  $E$ . 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

$$\min x(E)$$

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

$$x \geq 0$$

your instance:



$S$  e l'altro in  $V \setminus S$  (ossia gli archi del taglio di spiagge  $S$  e  $V \setminus S$ ).

### 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.

### Svolgimento esercizio 2 .

**Richiesta 1 (1 pt) (goal: one MST).**

Il cammino  $sABC$  ha costo  $2 + 1 + 2 = 5$ , parte dal nodo  $s$  ad arriva al nodo  $C$ .

**Richiesta 2 (1 pt) (goal: unicity).**

Indicate con  $d_A$ ,  $d_B$  e  $d_C$  i valori delle distanze dei nodi  $A$ ,  $B$  e  $C$  dal nodo  $s$  possiamo osservare che esse dovranno rispettare i vincoli del sistema di PL proposto con l'esercizio. Infatti, dal fatto che il grafo in figura ha un arco  $(A, C)$  di costo/lunghezza 4 disegua immediatamente che  $d_C \leq d_A + 4$  come ricordato nel quarto vincolo. Il terzo e il quinto vincolo hanno lo stesso significato, e provengono dall'arco  $(A, B)$  e dall'arco  $(B, C)$ , rispettivamente. Si è probabilmente preferito evitare di introdurre una variabile  $y_s$  da poi fissare con un vincolo  $y_s = 0$ , solo per questo il primo e il secondo vincolo appaiono avere una struttura leggermente diversa; è solo una questione di economia sul numero delle variabili e dei vincoli. Il problema di PL assegnato è in forma standard di massimizzazione, ad origine ammissibile, eppure nessuno dei tre nodi che hanno meritato una variabile è a distanza 0 da  $s$ . La verità è che ogni soluzione ammissibile  $x = (x_A, x_B, x_C)$  offre solamente un vettore di lower-bounds non-negativi sulle vere distanze (i vincoli di non-negatività sono stati convenientemente introdotti solo per garantire l'esistenza di una soluzione di base di partenza). Ma la funzione obiettivo tira la corda portando il valore del lower-bound  $d_C$  il più in alto possibile, di fatto al valore corretto della distanza di  $C$  da  $s$ . Nel suo sollevarsi come un palloncino di elio, in nodo  $C$  si tira dietro il nodo  $B$ , e quindi il nodo  $A$ , per rimanere infine inchiodati tutti e tre al nodo  $s$ : in questo caso, siccome il cammino minimo da  $s$  a  $C$  è un cammino Hamiltoniano, ne consegue che la soluzione ottima è una sola, dove ogni nodo andrà a collocarsi ad una quota che è la sua effettiva distanza dal nodo  $s$ .

La soluzione ottima richiesta è pertanto:  $\hat{x} = (\hat{x}_A, \hat{x}_B, \hat{x}_C)$ .

**Richiesta 3 (1 pt) (goal: explicit instance-specific).**

Il duale di un problema in forma standard si ottiene introducendo una variabile/moltiplicatore non-negativo per ogni vincolo. Alla luce di ciò abbiamo adottato un criterio naturale per il naming di queste variabili (a valle di questa scelta è tutto automatico e predeterminato)

$$\begin{aligned} \min \quad & 2y_{s,A} + 5y_{s,B} + y_{A,B} + 4y_{A,C} + 2y_{B,C} \\ & y_{s,A} - y_{A,B} - y_{A,C} \geq 0 \\ & y_{s,B} + y_{A,B} - y_{B,C} \geq 0 \\ & y_{A,C} + y_{B,C} \geq 1 \\ & y_{s,A}, y_{s,B}, y_{A,B}, y_{A,C}, y_{B,C} \geq 0 \end{aligned}$$

La soluzione  $\hat{x} = (2, 3, 5)$  soddisfa ad eguaglianza i vincoli 1, 3, 5, presentando invece variabili di slack non-nulle sui vincoli 2 e 4; quindi il primo gruppo di condizioni agli scarti complementari richiede che la seconda variabile duale ( $y_{s,B}$ ) e la quarta ( $y_{A,C}$ ) siano entrambe nulle (arco non teso non può esprimere reazione vincolare sul palloncino che intende salire). La soluzione non ha alcuna componente nulla, e quindi i vettori  $y$  che soddisfano il secondo set di condizioni degli scarti complementari devono soddisfare ad eguaglianza tutti e tre i vincoli del problema duale.

Con ciò perveniamo al sistema:

$$\begin{aligned} y_{s,A} - y_{A,B} &= 0 \\ +y_{A,B} - y_{B,C} &= 0 \\ +y_{B,C} &= 1 \end{aligned}$$

che prevede la sola soluzione (nell'ordine in cui i valori delle variabili vengono a determinarsi):

- $y_{s,B} = y_{A,B} = 0$  per primo set di condizioni agli scarti complementari
- $y_{B,C} = 1$  (l'arco/corda  $(B, C)$  è l'unico a tendersi per trattenere il palloncino  $C$  e tutta la spinta dell'elio di  $C$  si esprime attraverso  $B$ )
- $y_{A,B} = 1$  (l'arco/corda  $(A, B)$  è l'unico a tendersi per trattenere la spinta convogliata in  $B$  che si ritrova quindi interamente incanalata su  $A$ ).



- $y_{s,A} = 1$  (l'arco/corda  $(s, A)$  è l'unico a tendersi per trattenere la spinta convogliata in  $A$ , l'intera spinta dell'elio (il coefficiente 1 della nella funzione obiettivo del primale) si è ora scaricata a terra sul nodo  $s$  inchiodato al pavimento.

E' facile verificare che questa soluzione al sistema degli scarti elementari è soluzione ammissibile per il problema duale scritto sopra. Dopo questa verifica possiamo concludere che la soluzione primale  $\hat{x}$  che avevamo proposto e questa soluzione duale sono entrambe ottime per il corollario del teorema della dualità debole in quanto attengono lo stesso valore nelle rispettive funzioni obiettivo.

Siccome i tre moltiplicatori  $y_{B,C} = 1$ ,  $y_{A,B} = 1$  e  $y_{s,A} = 1$  sono tutti non nulli allora in ogni soluzione ottima primale gli slack associati a questi archi devono essere nulli, il che conduce a ricostruire univocamente la soluzione  $\hat{x}$  che è pertanto l'unica soluzione ottima. Equivalentemente: quando una soluzione di base ammissibile presenta tutti i costi ridotti negativi è ottima, e se sono tutti strettamente negativi l'ottimo è unico.

La soluzione duale ottima è unica in quanto la soluzione primale è unica e non degenera.

#### Richiesta 4 (1 pt) (goal: canonic form).

Per ottenere la forma canonica e/o il primo dizionario il primo passo consiste nell'introduzione/definizione/naming delle variabili di slack. In questo caso (essendo il problema in forma standard) avremo una variabile di slack per ciascun vincolo. Anche in coerenza col naming per le variabili duali,  $s_{s,A}$ ,  $s_{s,B}$ ,  $s_{A,B}$ ,  $s_{A,C}$ ,  $s_{B,B}$  potrebbe essere una buona scelta per i nomi della variabili di slack.

Spesa questa scelta, la forma canonica e primo dizionario sono:

DIZIONARIO DI FORMA CANONICA

$$\begin{cases} z = 0 + 0x_A + 0x_B + 1x_C \\ s_A = 2 - 1x_A + 0x_B + 0x_C \\ s_B = 5 + 0x_A - 1x_B + 0x_C \\ s_{A,B} = 1 + 1x_A - 1x_B + 0x_C \\ s_{A,C} = 4 + 1x_A + 0x_B - 1x_C \\ s_{B,C} = 2 + 0x_A + 1x_B - 1x_C \end{cases}$$

max  $z$  :

$$x_A, x_B, x_C, s_A, s_B, s_{A,B} \geq 0$$

$$s_{A,C}, s_{B,C} \geq 0$$

TABLEAU DI FORMA CANONICA

	-	$x_A$	$x_B$	$x_C$
$z$	0	0	0	1
$s_A$	2	-1	0	0
$s_B$	5	0	-1	0
$s_{A,B}$	1	1	-1	0
$s_{A,C}$	4	1	0	-1
$s_{B,C}$	2	0	1	(-1)

#### Richiesta 5 (1 pt) (goal: auxiliary problem).

Il primo pivot è obbligato (e lo abbiamo già indicato anche nel tableau): l'unica variabile spinta ad entrare in base dalla funzione obiettivo è la  $x_C$ , e la variabile ad uscire per prima è di necessità la  $s_{B,C}$  poichè l'arco  $(B, C)$  è il primo a tendersi cominciando a tirare sul nodo  $B$  non appena il palloncino di elio (il nodo  $C$ ) non avrà raggiunto quota  $z = 2$ .

Possiamo prevedere anche i pivot successivi: il nodo  $B$  sarà il prossimo a sollevarsi dal pavimento (ossia la prossima variabile ad assumere valore non-nullo ed entrare in base) e verrà arrestato dal nodo  $A$  per esaurimento dello slack  $s_{A,B}$  concesso dall'arco/corda  $(A, B)$ . Sarà quindi  $s_{A,B}$  la variabile uscente al secondo pivot. Al terzo ed ultimo pivot entrerà  $x_A$  ed uscirà  $s_A$ . A questo punto alle tre corde/archi tesi corrisponderanno le variabili in fuori-base  $s_A = s_{A,B} = s_{B,C} = 0$  con variabili in base  $(x_A, x_B, x_C, s_B, s_{A,C}) = (2, 3, 5, 2, 1)$ . I primi tre valori essendo le distanze dei nodi dal nodo  $s$  e gli ultimi due valori lo slack delle due corde non tese.

#### Richiesta 6 (1 pt) (goal: first tableau).

Effettuando il primo pivot, si perviene alle seguenti rappresentazioni della seconda soluzione di base:

### DIZIONARIO DI SOL BASE 2

$$\begin{cases} z = 2+0x_A+1x_B-1s_{B,C} \\ s_A = 2-1x_A+0s_{B,C} \\ s_B = 5+0x_A-1x_B+0s_{B,C} \\ s_{A,B} = 1+1x_A-1x_B+0s_{B,C} \\ s_{A,C} = 2+1x_A-1x_B+1s_{B,C} \\ x_C = 2+0x_A+1x_B-1s_{B,C} \end{cases}$$

max  $z$  :

$$x_A, x_B, s_{B,C}, s_A, s_B, s_{A,B} \geq 0$$

$$s_{A,C}, x_C \geq 0$$

### TABLEAU DI SOL BASE 2

	-	$x_A$	$x_B$	$s_{B,C}$
$z$	2	0	1	-1
$s_A$	2	-1	0	0
$s_B$	5	0	-1	0
$s_{A,B}$	1	1	(-1)	0
$s_{A,C}$	2	1	-1	1
$x_C$	2	0	1	-1

La soluzione di base associata a queste scritte non è ottima data la presenza di un coefficiente strettamente positivo (lo 1) nell'espressione attuale della funzione obiettivo. Nel prossimo pivot la variabile entrante è la  $x_B$  (forzata, il coefficiente positivo è uno solo) e quella uscente è la  $s_{A,B}$  (forzata, questa è la corda che si strapperebbe prima di ogni altra all'aumentare del valore della  $x_B$  senza pivotare).

Effettuando il secondo pivot, si perviene alle seguenti rappresentazioni della terza soluzione di base che andiamo a visitare:

### DIZIONARIO DI SOL BASE 3

$$\begin{cases} z = 3+1x_A-1s_{A,B}-1s_{B,C} \\ s_A = 2-1x_A+0s_{A,B}+0s_{B,C} \\ s_B = 4-1x_A+1s_{A,B}+0s_{B,C} \\ x_B = 1+1x_A-1s_{A,B}+0s_{B,C} \\ s_{A,C} = 1+0x_A+1s_{A,B}+1s_{B,C} \\ x_C = 3+1x_A-1s_{A,B}-1s_{B,C} \end{cases}$$

max  $z$  :

$$x_A, s_{A,B}, s_{B,C}, s_A, s_B, x_B \geq 0$$

$$s_{A,C}, x_C \geq 0$$

### TABLEAU DI SOL BASE 3

	-	$x_A$	$s_{A,B}$	$s_{B,C}$
$z$	3	1	-1	-1
$s_A$	2	(-1)	0	0
$s_B$	4	-1	1	0
$x_B$	1	1	-1	0
$s_{A,C}$	1	0	1	1
$x_C$	3	1	-1	-1

La soluzione di base associata a queste scritte non è ottima data la presenza di un coefficiente strettamente positivo (lo 1) nell'espressione attuale della funzione obiettivo. Nel prossimo pivot la variabile entrante è la  $x_A$  (forzata, il coefficiente positivo è uno solo) e quella uscente è la  $s_A$  (forzata, questa è la corda che si strapperebbe prima di ogni altra all'aumentare del valore della  $x_A$  senza pivotare).

Effettuando il terzo pivot, si perviene alle seguenti rappresentazioni della quarta ed ultima soluzione di base che viene visitata dal metodo del simplesso:

### DIZIONARIO DI SOL BASE 4

$$\begin{cases} z = 5-1s_A-1s_{A,B}-1s_{B,C} \\ x_A = 2-1s_A+0s_{A,B}+0s_{B,C} \\ s_B = 2+1s_A+1s_{A,B}+0s_{B,C} \\ x_B = 3-1s_A-1s_{A,B}+0s_{B,C} \\ s_{A,C} = 1+0s_A+1s_{A,B}+1s_{B,C} \\ x_C = 5-1s_A-1s_{A,B}-1s_{B,C} \end{cases}$$

max  $z$  :

$$s_A, s_{A,B}, s_{B,C}, x_A, s_B, x_B \geq 0$$

$$s_{A,C}, x_C \geq 0$$

### TABLEAU DI SOL BASE 4

	-	$s_A$	$s_{A,B}$	$s_{B,C}$
$z$	5	-1	-1	-1
$x_A$	2	-1	0	0
$s_B$	2	1	1	0
$x_B$	3	-1	-1	0
$s_{A,C}$	1	0	1	1
$x_C$	5	-1	-1	-1

La soluzione di base associata a queste scritture è ottima data l'assenza di coefficienti strettamente positivi nell'espressione attuale della funzione obiettivo. Nel dizionario e tableau attuale è immediato trovare riscontro della soluzione ottima già anticipata in richieste precedenti.

Come prova del nove conviene a questo punto introdurre la soluzione di base ottima testè computata  $(x_A, x_B, x_C, s_A, s_B, s_{A,B}, s_{A,C}, s_{B,C}, z) = (2, 3, 5, 2, 1, 5)$  nel primissimo dizionario, controllando che ogni equazione ne risulti soddisfatta.

**Richiesta 7 (6 pt) (goal: first phase).**

Non vi è alcun interesse nell'allungare le corde/archi  $(s, B)$  e  $(A, C)$  dato che  $s_B = 2 > 0$  e  $s_{A,C} = 1 > 0$ . Leggendo invece i costi ridotti (coefficienti delle variabili nell'espressione della funzione obiettivo  $z$ ) nell'ultimo dizionario riscontriamo che si sarebbe disposti a pagari fino ad 1 unità per allungare uno qualsiasi degli archi  $(s, A)$ ,  $(A, B)$  e  $(B, C)$ . Infatti, se uno qualsiasi di questi archi venisse allungato di 1, la distanza di  $C$  da  $s$  crescerebbe di uno e di tanto avanzerebbe il palloncino di elio. Chi fosse interessato a minimizzare la distanza di  $C$  sarebbe disposto a pagare gli stessi identici valori, in fondo ciò è dovuto al fatto che la soluzione ottima è non-degenere. Nel grafo di due soli nodi  $s$  e  $t$  e di due archi entrambi con coda in  $s$ , testa in  $t$  e lunghezza 3, la distanza di  $t$  da  $s$  è 3 e il palloncino di elio non sarebbe disposto a pagare nemmeno 0,01 (e nemmeno  $\varepsilon$  per nessun  $\varepsilon > 0$ ) per allungare uno solo dei due archi che simultaneamente trattengono  $t$ . Invece chi fosse interessato a ridurre la distanza di  $t$  da  $s$  sarebbe disposto a pagare fino ad 1. Tornando all'istanza del testo, il prezzo ombra dell'arco  $(s, A)$  è 1 ma solo per incrementi  $\delta \leq 2$  della sua lunghezza. Il prezzo ombra dell'arco  $(A, B)$  è 1 ma solo  $\delta \leq 1$ . Il prezzo ombra dell'arco  $(B, C)$  è 1 ma solo  $\delta \leq 1$ .

**Richiesta 8 (2 pt) (goal: change horse).**

Una volta dati i nomi alle variabili di pertinenza del duale (abbiamo già introdotto i nomi della variabili di decisione  $y$  e stabilita ora la lettera  $w$  per le variabili di surplus, con la scelta di rimanere anche per esse coerenti con quanto nel primale per quanto riguarda i pedici) si tratta solo di trasporre il tableau che esprime la soluzione primale ottima (avendo cura di invertire i segni tranne che per i numeri che passano dal costituire la prima riga al costituire la prima colonna). Si otterrà quindi:

DIZIONARIO DI SOL OTTIMA DUALE

$$\begin{cases} z = 5 + 2w_A + 2y_B + 3w_B + 1y_{A,C} + 5w_C \\ y_A = 1 - 1w_A + 1y_B - 1w_B + 0y_{A,C} - 1w_C \\ y_{A,B} = 1 + 0w_A + 1y_B - 1w_B + 1y_{A,C} - 1w_C \\ y_{B,C} = 1 + 0w_A + 0y_B + 0w_B + 1y_{A,C} - 1w_C \end{cases}$$

max  $z$  :

$$w_A, y_B, w_B, y_{A,C}, w_C, y_A \geq 0$$

$$y_{A,B}, y_{B,C} \geq 0$$

TABLEAU DI SOL OTTIMA DUALE

	-	$w_A$	$y_B$	$w_B$	$y_{A,C}$	$w_C$
$z$	5	2	2	3	1	5
$y_A$	1	-1	1	-1	0	-1
$y_{A,B}$	1	0	1	-1	1	-1
$y_{B,C}$	1	0	0	0	1	-1

La soluzione ottima duale estesa è:  $(y_A, y_B, y_{A,B}, y_{A,C}, y_{B,C}, w_A, w_B, w_C, z) = (1, 0, 1, 0, 1, 0, 0, 0, 5)$ . In pratica è presente una tensione di valore 1 su ciascuna delle tre corde tese. Invece ciascuno dei tre nodi è in equilibrio statico ( $w_A = 0, w_B = 0, w_C = 0$ ).

**Richiesta 9 (1 pt) (goal: second phase).**

Indichiamo con  $A_s$  l'insieme degli archi con coda in  $s$  e con  $A_{\bar{s}}$  tutti gli altri archi. Introdotta una variabile  $x_v$  per ogni nodo  $v \in V \setminus \{s\}$ , il modello generale sarebbe

$$\begin{aligned}
\max z &= x_t \\
x_v &\leq c_{s,v} \text{ per ogni arco } (s,v) \in A_s \\
x_v - x_u &\leq c_{u,v} \text{ per ogni arco } (u,v) \in A_{\bar{s}} \\
x &\geq 0 \text{ per ogni nodo } v \in V
\end{aligned}$$

Si noti come ogni soluzione ammissibile  $x$  esprima dei lower-bound sulle distanze da  $s$ . Se questo problema risultasse unbounded ne posso quindi dedurre che nel grafo non è presente alcun cammino da  $s$  a  $t$ . In caso contrario, il valore ottimo  $z = x_t$  è proprio la distanza di  $t$  da  $s$  e resta da svelare solo come ricostruire un cammino di tale costo/lunghezza. Lo ricostruisco a ritroso, partendo con  $v = t$  e per applicazione ricorsiva del seguente passo: se  $v = s$  allora il processo di ricostruzione è finito, altrimenti scelgo come nodo precedente a  $v$  nel cammino in costruzione un qualsiasi nodo  $u \in V$  tale che  $x_v = x_u + c_{u,v}$ ; si noti che un tale nodo deve esistere per l'invariante di ciclo che il lower-bound  $x_v$  è la vera distanza di  $v$  da  $s$ . Ne posso inoltre dedurre che il lower-bound  $x_u$  è la vera distanza di  $u$  da  $s$ .

Come già anticipato sopra, il problema sarebbe unbounded se e solo se il nodo  $t$  non fosse raggiungibile dal nodo  $s$ . Se bounded esiste una soluzione ottima dato che il problema è in forma standard ad origine ammissibile, ed essa sarebbe unica se e solo se il cammino ottimo da  $s$  a  $t$  dovesse essere un cammino Hamiltoniano. Una soluzione duale ottima esiste precisamente quando esiste una soluzione primale ottima; è poi unica se e solo se la soluzione primale ottima non è degenere ossia se e solo se nella ricostruzione a ritroso del cammino non mi ritrovo mai con l'imbarazzo nella scelta del nodo  $u$  a precedere il nodo  $v$ .

**Richiesta 10 (2 pt) (goal: dual).**

Indichiamo con  $A_s$  l'insieme degli archi con coda in  $s$  e con  $A_{\bar{s}}$  tutti gli altri archi. Introdotta una variabile  $d_v$  per ogni nodo  $v \in V \setminus \{s\}$ , consideriamo il seguente modello generale

$$\begin{aligned}
\max z &= \sum_{v \in V \setminus \{s\}} d_v \\
d_v &\leq c_{s,v} \text{ per ogni arco } (s,v) \in A_s \\
d_v - d_u &\leq c_{u,v} \text{ per ogni arco } (u,v) \in A_{\bar{s}} \\
d &\geq 0 \text{ per ogni nodo } v \in V \setminus \{s\}
\end{aligned}$$

Ogni soluzione ammissibile  $x$  esprime dei lower-bound sulle distanze da  $s$ . Questo problema risulta unbounded se e solo se un qualche nodo non è raggiungibile da  $s$ . In caso contrario, la soluzione ottima è sempre unica e si ha con  $d_v$  pari al valore della distanza di  $v$  da  $s$  per ogni nodo  $v \in V \setminus \{s\}$ . Resta da svelare solo come ricostruire un albero dei cammini minimi: ciascun nodo  $v \in V \setminus \{s\}$  scelga come proprio padre un qualsiasi nodo  $u \in V$  tale che  $d_v = d_u + c_{u,v}$ ; si noti che un tale nodo deve esistere per l'ottimalità della soluzione.

**Richiesta 11 (1 pt) (goal: dual opts).**

**Lemma:** Ogni prefisso di cammino minimo è cammino minimo. **proof:** Sia  $P_v$  is prefisso fino ad un qualche nodo  $v$  di un qualche cammino minimo  $\bar{P}_z$  da  $s$  a  $z$ . Sia  $P_{v,z}$  il suffisso di  $P_z$  che parte da  $v$  e sia  $\bar{P}_v$  un cammino di costo minimo da  $s$  a  $v$ . Allora  $\bar{P}_v + P_{v,z}$  contiene un cammino da  $s$  a  $v$  di costo non inferiore a quello del cammino ottimo  $\bar{P}_z$  e quindi il costo del cammino ottimo  $\bar{P}_v$  è non inferiore a quello di  $P_v$  che sarà pertanto altrettanto ottimo. QED

**grafi non-diretti:** nel triangolo di nodi  $s, A, B$  con  $c_{sA} = c_{sB} = 1$  e  $c_{AB} = -1$  l'unico cammino di costo minimo da  $s$  a  $A$  è  $sBA$  di costo  $1 + (-1) = 0$ ; e l'unico cammino di costo minimo da  $s$  a

$B$  è  $sAB$  di costo  $1 + (-1) = 0$ . Un sottografo che contenga tutti gli archi di tutti i cammini minimi dovrebbe contenere quindi tutti e tre gli archi del triangolo e quindi un ciclo - non potrebbe essere quindi nè una foresta nè un albero.

**grafi diretti:** si adatti l'esempio di cui sopra come segue: si orienti l'arco  $sA$  da  $s$  ad  $A$  mantenendo il suo costo a 1, si orienti l'arco  $sB$  da  $s$  ad  $B$  mantenendo il suo costo a 1, si sostituisca l'arco  $AB$  da  $A$  ad  $B$  coi due archi diretti  $(A, B)$  e  $(B, A)$  entrambi di costo  $-1$ .

**Richiesta 12 (6 pt) (goal: complementary slackness).**

Per esprimere un'istanza  $G = (V, E)$  di HAMILTONIAN CYCLE come un'istanza di HAMILTONIAN S,T-PATH basta, assumendo che  $G$  sia un grafo diretto, scegliere un qualsiasi nodo  $v$  in  $V$  ed aprirlo in due nodi  $v^-$  e  $v^+$  avendo cura di fare entrare in  $v^-$  (uscire da  $v^+$ ) ogni arco che prima entrava in  $v$  (usciva da  $v$ ). Ovviamente, ogni ciclo Hamiltoniano in  $G$  è un  $v^+, v^-$ -path Hamiltoniano nel grafo così ottenuto e, di converso, un qualsiasi  $v^+, v^-$ -path Hamiltoniano nel grafo così ottenuto può essere riguardato come un ciclo Hamiltoniano in  $G$ .

Il prossimo passo è lo stesso sia che  $G$  sia un grafo diretto che ove  $G$  fosse invece un grafo non-diretto. Per esprimere un'istanza  $(G = (V, E), s, t \in V)$  di HAMILTONIAN S,T-PATH come un'istanza di SHORTEST S,T-PATH basta, assumendo che  $G$  sia un grafo diretto, assegnare costo  $-1$  a ciascuno dei suoi archi. Sono immediati i due lemmi che confermano la correttezza di tale riduzione:

**Lemma Easy:** Se  $G$  ammette un  $s, t$  ---path Hamiltoniano  $P$  allora  $P$  è un  $s, t$  ---path di costo  $1 - |V|$  in  $(G, c)$ .

**Lemma Hard:** Se  $(G, c)$  ammette un  $s, t$  ---path  $P$  di costo al più  $1 - |V|$  allora il costo di  $P$  è precisamente  $1 - |V|$  e  $P$  è un  $s, t$  ---path Hamiltoniano in  $G$ .

Per ottenere l'ultimo risultato richiesto potremmo partire dalla conoscenza assodata che HAMILTONIAN S,T-PATH è noto essere NP-completo anche per grafi non-diretti.

**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?

### Svolgimento esercizio 3 .

#### Richiesta 1

Per rispondere alle prime richieste conviene compilare preventivamente un paio di tabelle di programmazione dinamica, di fatto quelle oggetto delle prime due richieste.

Ecco la prima tabella esplicitamente richiesta (max\_len\_last\_at, compilata da sinistra):

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
1	1	2	3	4	2	3	4	5	3	4	6	3	5	4	6	7	3	4	2	5	7	5	6	8	6	5
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

#### Richiesta 2

Ed ecco la seconda (max\_len\_first\_at), compilata da destra (la nuova riga subito sotto a quella coi valori della sequenza in input):

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
1	1	2	3	4	2	3	4	5	3	4	6	3	5	4	6	7	3	4	2	5	7	5	6	8	6	5
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
6	8	7	4	1	7	6	5	4	6	5	1	5	4	4	3	1	5	4	4	3	2	3	2	1	1	1

Il fatto che il massimo valore sia 8 sia nella riga prodotta per evadere la Richiesta 1 che nella riga ora inserita per evadere la Richiesta 2 viene a verifica parziale della correttezza dei conteggi.

### Richiesta 3

La massima lunghezza di una sottosequenza strettamente crescente di  $S$  è pertanto appunto 8. Volendo ricostruire una sottosequenza strettamente crescente di tale lunghezza si procede a ritroso rispetto all'ordine in cui i numeri nella riga (possiamo lavorare su quella prodotto nella Richiesta 1, oppure su quella prodotto nella Richiesta 2, è in differente) sono stati calcolati, ed utilizzarli come angioletti compagnetti (oracoli che ci anticipano le conseguenze di ogni nostra scelta, se includere o meno l'elemento nella posizione corrente del nostro esodo). Affidandoci a tali grilli parlanti diviene facile affrontare a colpo sicuro ogni scelta guidati dalla forza della promessa.

Le soluzioni ottime di un problema combinatorico come questo possono essere anche in numero esponenziale, ma eccone una:

<b>pos</b>	<b>2</b>	<b>3</b>	<b>7</b>	<b>8</b>	<b>9</b>	<b>16</b>	<b>22</b>	<b>25</b>
$S[i]$	10	19	28	39	45	55	56	57

E' facile verificare l'ammissibilità di tale soluzione, ma come verificarne l'ottimalità?

### Richiesta 4

Poichè una sottosequenza strettamente crescente ed una sottosequenza mai crescente non possono condividere più di un singolo elemento della sequenza di riferimento, allora un set di  $k$  sottosequenze mai crescenti che coprano ogni elemento della sequenza di riferimento dimostrano che nessuna sottosequenza strettamente crescente può contenere più di  $k$  elementi (contenendone al più uno per ogni sottosequenza mai crescente).

Ecco quindi il nostro certificato di ottimalità della soluzione fornita al punto precedente, nella forma di  $k = 8$  sottosequenze mai crescenti che coprano ogni elemento della sequenza  $S$  di riferimento per questo esercizio:

<b>pos</b>	<b>1</b>	<b>2</b>
$S[i]$	31	10

<b>pos</b>	<b>3</b>	<b>6</b>	<b>20</b>
$S[i]$	19	17	14

<b>pos</b>	<b>4</b>	<b>7</b>	<b>10</b>	<b>13</b>	<b>18</b>
$S[i]$	50	28	26	21	20

<b>pos</b>	<b>5</b>	<b>8</b>	<b>11</b>	<b>15</b>	<b>19</b>
$S[i]$	60	39	36	30	22

<b>pos</b>	<b>9</b>	<b>14</b>	<b>21</b>	<b>23</b>	<b>27</b>
$S[i]$	45	42	37	24	23

<b>pos</b>	<b>12</b>	<b>16</b>	<b>24</b>	<b>26</b>
$S[i]$	59	55	47	40

<b>pos</b>	<b>17</b>	<b>22</b>
$S[i]$	57	56

<b>pos</b>	<b>25</b>
$S[i]$	57

### Richiesta 5

La massima lunghezza di una sottosequenza strettamente crescente di  $S$  tra quelle che terminano con l'elemento in posizione 19 è il 4 che si trova nella colonna 19 della riga di programmazione dinamica `last_in_pos` introdotta con la Richiesta 1. La tecnica/mantra per ricostruire una soluzione ottima è quella di consultare ad ogni scelta la tabella di programmazione dinamica, come spiegato sopra. La parola di Javè non è univoca ma potente in ricchezza generativa: le soluzioni ottime di un problema combinatorico come questo possono essere anche in numero esponenziale. Per soddisfare la richiesta ci basta esibirne una:

<b>pos</b>	<b>2</b>	<b>6</b>	<b>18</b>	<b>19</b>
$S[i]$	10	17	20	22

### Richiesta 6

La massima lunghezza di una sottosequenza strettamente crescente di  $S$  tra quelle che di  $S$  non prendono alcun elemento di indice inferiore a 8 è 6 come si può evincere dalla riga di programmazione dinamica `first_in_pos` introdotta con la Richiesta 2. (Volendo rendere questa riga più leggibile, si potrebbe aggiungere come ulteriore riga di programmazione dinamica il calcolo del massimo valore che lei offre a destra di ciascuna posizione, poi consultare tale riga nella colonna 8.) La tecnica/mantra per ricostruire una soluzione ottima è sempre quella (anche per altri problemi di ottimizzazione risolti tramite programmazione dinamica). Per soddisfare la richiesta ci basta esibire una soluzione ottima:

<b>pos</b>	<b>10</b>	<b>11</b>	<b>14</b>	<b>16</b>	<b>22</b>	<b>25</b>
$S[i]$	26	36	42	55	56	57

### Richiesta 7

Combinando il dato in colonna 13 dalle due tabelle compilate per le Richieste 1 e 2 scopriamo che la massima lunghezza di una sottosequenza strettamente crescente di  $S$  che prenda l'elemento in posizione 13 è  $\mathbf{3 + 5 - 1 = 7}$ . Ricostruendo al rispettivo ritroso entrambe (verso destra oppure verso sinistra) le ali di una soluzione ottima si ottiene:

<b>pos</b>	<b>2</b>	<b>6</b>	<b>13</b>	<b>14</b>	<b>16</b>	<b>22</b>	<b>25</b>
$S[i]$	10	17	21	42	55	56	57

### Richiesta 8

Per trovare un punto di ripensamento favorevole per la N-sottosequenza di  $S$  ho confrontato i valori nelle tabelle compilate per rispondere alle Richieste 1 e 2. Per semplificare tale confronto (da quadratico a lineare) ho preferito introdurre due ulteriori righe di PD (una, compilata da sinistra, riporta la massima lunghezza di una sottosequenza strettamente crescente per ogni prefisso; l'altra, compilata da destra, riporta la massima lunghezza di una sottosequenza strettamente crescente per ogni suffisso; sono ovvie le ricorrenze con cui calcolarle avvalendosi dei valori già prodotti per rispondere alle Richieste 1 e 2).

[illegible]



1	1	2	3	4	2	3	4	5	3	4	6	3	5	4	6	7	3	4	2	5	7	5	6	8	6	5
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
6	8	7	4	1	7	6	5	4	6	5	1	5	4	4	3	1	5	4	4	3	2	3	2	1	1	1
8	8	7	7	7	7	6	6	6	6	5	5	5	5	5	5	5	5	4	4	3	3	3	2	1	1	1

In fondo ogni N-sottosequenza può essere spezzata in due sottosequenze strettamente crescenti, una sinistra ed una destra. La tabella qui sopra può essere compilata in tempo lineare e consente di stabilire in tempo lineare che possiamo far terminare in posizione 17 la sottosequenza sinistra e far partire in posizione 18 la sottosequenza destra.

Procedendo a ritroso sulle due ali come da competenza già esibita otteniamo la seguente N-sequenza di massima lunghezza:

pos	2	6	10	11	14	16	17	18	19	21	22	25
$S[i]$	10	17	26	36	42	55	57	20	22	37	56	57

Essa è lunga **12**.

### Richiesta 9

Dalla seguente tabella evinco che sono **8** le sottosequenze strettamente crescenti di  $S$  di massima lunghezza. La riga centrale della tabella riporta la sequenza in input, mentre le due righe collocate sopra (sotto) di essa sono compilate da sinistra (da destra). Le righe limitrofe alla sequenza in input (max\_len\_last\_at e max\_len\_first\_at) le abbiamo già incontrate in precedenti tabelle ma vengono qui riprese in quanto ancillari al computo delle due ulteriori righe che ci servono ora (num\_opts\_last\_at e num\_opts\_first\_at). L'idea è che in ogni posizione  $i$ , con  $1 \leq i \leq n$ , la riga num\_opts\_last\_at (num\_opts\_first\_at) riporti il numero delle più lunghe sottosequenze strettamente crescenti di  $S$  che di  $S$  prendano l'elemento in posizione  $i$  come loro ultimo (primo) elemento.

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
1	1	1	1	1	1	2	2	2	2	4	2	2	6	6	8	8	2	4	1	14	8	4	26	8	18	4
1	1	2	3	4	2	3	4	5	3	4	6	3	5	4	6	7	3	4	2	5	7	5	6	8	6	5
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
6	8	7	4	1	7	6	5	4	6	5	1	5	4	4	3	1	5	4	4	3	2	3	2	1	1	1
3	8	4	1	1	4	3	2	1	1	1	1	7	1	3	1	1	3	3	3	2	1	1	1	1	1	1

Anche qui ho preferito compilare sia delle righe di PD (programmazione dinamica) da sinistra (che rispondono a domande per ogni prefisso di  $S$ ) che delle righe di PD da destra (che rispondono alle domande speculari per ogni suffisso di  $S$ ) in modo da trovare verifica almeno parziale nel confronto dei valori ottenuti per i due (quello sinistro e quello destro) sotto-problemi più grossi, ultimi a cadere come birilli sotto i colpi della ovvia ricorrenza che definisce la rispettiva famiglia. Per altro questo doppio conteggio mi serve se poi voglio rispondere in economia a tutte le richieste che seguono.

### Richiesta 10

Sono **4** le sottosequenze strettamente crescenti di  $S$  di massima lunghezza tra quelle che prendono l'elemento in posizione 19 come loro ultimo elemento.

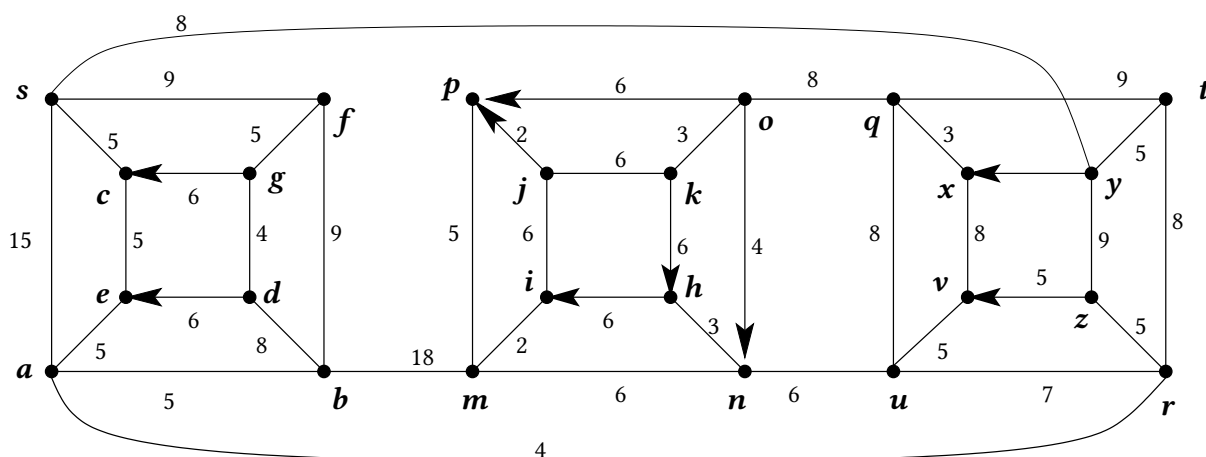
### Richiesta 11

Sono **2** le sottosequenze strettamente crescenti di  $S$  di massima lunghezza tra quelle che prendono l'elemento in posizione 8 come loro primo elemento.

**Richiesta 12**

Sono  $14 = 2 \times 7$  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]):



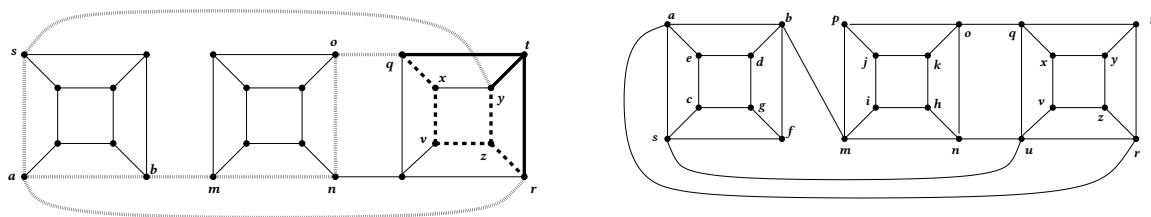
#### Richieste dell'Esercizio 4

- 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)

#### Svolgimento esercizio 4 .

##### Richiesta 1

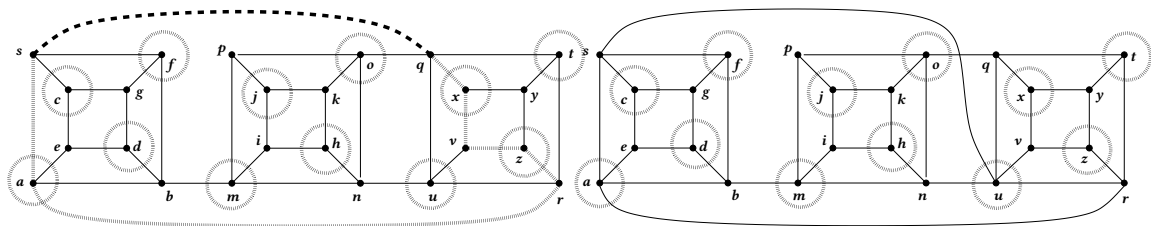
La non-planarit  di  $G$    certificata dalla  $K_{3,3}$  subdivision in figura, sulla sinistra. Sulla destra, si offre un planar embedding di  $G_u$ , che ne certifica invece la planarit .



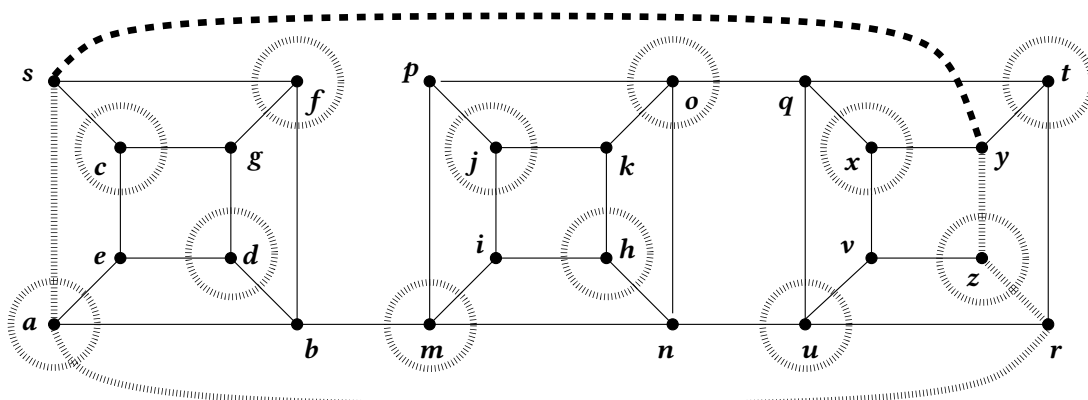
Un planar embedding di  $G_q$  è invece fornito nella figura a seguire, sulla sinistra.

## Richiesta 2

Nella figura a seguire, sulla sinistra, si mostra anche come il grafo  $G_q$  possa essere reso bipartito rimuovendo un solo arco (oltre alla bipartizione è evidenziato un ciclo dispari che certifica che almeno un arco deve essere rimosso). Sulla destra possiamo verificare che  $G_u$  era invece bipartito già di suo.



Su  $G$  serve rimuovere almeno un arco, e un solo arco di nuovo basta, ogni certificato richiesto è evidenziato in figura.



## Richieste 3 e 4

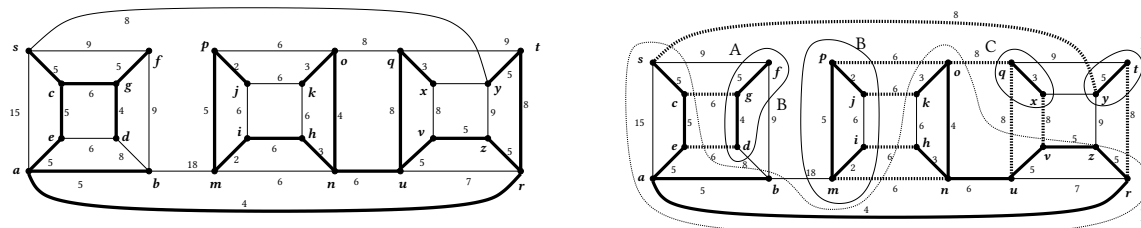
Un flusso ottimo (valore 16) è visualizzato in figura, con l'unico  $s, t$ -taglio minimo e dello stesso valore: quello che separa i nodi sul lato sinistro da quelli sul lato destro della figura (precisamente a metà, tagliando nel mezzo anche il cubo centrale). Questo taglio da solo certifica l'ottimalità del flusso in figura, come di ogni altro possibile flusso di valore 16 (ce ne sono diversi).



**linea spessa continua** quelli che appartengono ad ogni MST

**linea spessa tratteggiata** quelli che appartengono a qualche MST ma non a tutti

**linea sottile** quelli che non appartengono ad alcun MST



### Richiesta 9

Gli archi in linea spessa raccolgono i nodi in 5 isole  $A$ ,  $B$ ,  $C$ ,  $D$  ed  $E$ .

Ogni MST dovrà prendere almeno un arco del taglio che separa la componente  $A$  dal resto del grafo. Ogni tale arco ha un estremo in  $A$  e l'altro in  $E$ , pertanto la questione si riduce a scegliere esattamente uno di questi archi sapendo che sono intercambiabili. Converrà prendere un arco di peso minimo (6). Abbiamo due opzioni: arco  $cg$  oppure arco  $ed$ . In modo del tutto analogo, per collegare la componente  $B$  al resto del grafo andrà preso uno ed un solo arco di peso minimo con un estremo nella componente  $B$  e l'altro fuori, prendere più di un arco con un estremo in  $B$  non serve a nulla. Quindi questa scelta è tra 4 opzioni, e del tutto indipendente dalla scelta precedente. In modo del tutto analogo, per saldare la componente  $C$  al blocco  $E + A + B$  dobbiamo scegliere un arco di peso 8 (ci sono 3 opzioni). Infine, per saldare la componente  $D$  al blocco  $E + A + B + C$  dobbiamo scegliere un arco di peso 8 (di nuovo 3 opzioni).

Il numero totale di alberi ricoprenti di peso minimo è pertanto  $2 \times 4 \times 3 \times 3 = 72$ .

### Richiesta 10

Forniamo ora dei certificati specifici per la classificazione dei 4 archi per cui richiesto:

**arco  $mb$**  in nessun MST in quanto arco di peso strettamente massimo nel ciclo  $mnurab$ .

**arco  $mp$**  in tutti gli MST in quanto arco di peso strettamente minimo del taglio di spiaggia  $\{m, i\}$  (ossia del taglio che separa i nodi  $m$  ed  $i$  dal resto del grafo).

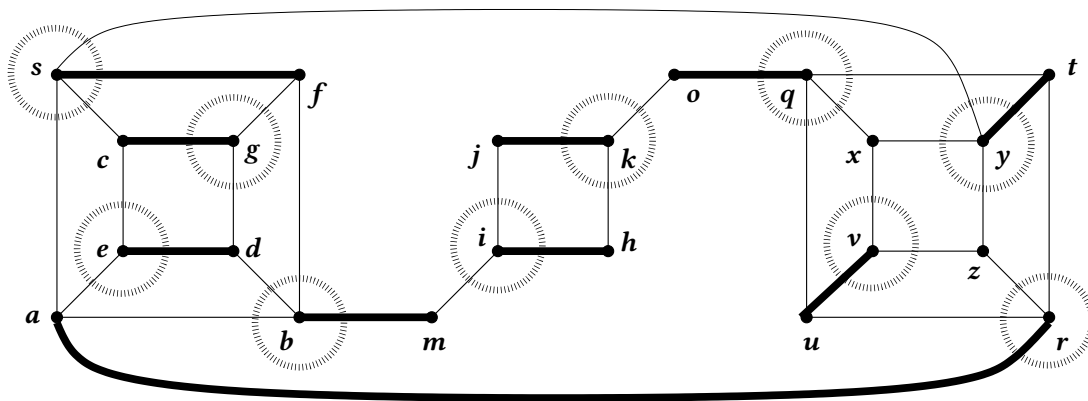
**arco  $mi$**  in tutti gli MST in quanto arco di peso strettamente minimo del taglio di spiaggia  $m$  (ossia del taglio che separa  $m$  dal resto del grafo).

**arco  $mn$**  in qualche MST in quanto arco di peso minimo del taglio che separa la componente  $D$  dal resto del grafo. Non in tutti in quanto arco di peso massimo nel ciclo  $mnop$ .

**arco  $qu$**  in qualche MST in quanto arco di peso minimo del taglio che separa la componente  $B$  dal resto del grafo. Non in tutte in quanto arco di peso massimo nel ciclo  $quno$ .

### Richiesta 11

Nel grafo  $G_{rs}$  evidenziamo un matching  $M$  ad un node cover  $X$  con  $|X| = |M|$ , che mutualmente certificano la propria ottimalità dato che in nessun grafo possono aversi un matching  $M$  ed un node cover  $X$  con  $|X| < |M|$  (di ogni arco di un matching ogni node cover deve prendere almeno uno dei due estremi). Siccome il grafo era bipartito era di fatto garantito che una tale coppia di oggetti dovesse esistere.



## CONSIGLI SU COME PREPARARSI ALL'ESAME

Per conseguire un voto per l'insegnamento di Ricerca Operativa devi partecipare ad un appello di esame. Il primo appello d'esame di ogni anno accademico ha luogo a giugno, dopo la conclusione del corso. Per gli appelli estivi in aula delta, non abbiamo controllo dell'aria condizionata e l'ambiente potrà risultarvi troppo freddo. Data la durata dell'appello consiglio di portarsi golfini, snack, acqua e matite o pennarelli colorati. Potete portarvi materiali cartacei ma non è consentita alcuna strumentazione elettronica. Dovete portare il tesserino col vostro numero di matricola.

Durante l'esame, dovrete lavorare per almeno 4 ore a quella che definisco «una prova di cromatografia su carta». Serve per riconoscervi con ragionevole confidenza quanto avete lavorato, appreso, sedimentato. E trasformare questo in una proposta di voto la più congrua possibile. La logica dello svolgimento dell'esame deve essere quella di dimostrare al meglio le competenze acquisite andando con efficienza a raccogliere, dei molti punti messi in palio a vario titolo: cercate e concretizzate quelli che più vi convengono, non impegolatevi a dimostrare quello che non sapete o dove incontrate incertezze. Contano le risposte corrette, fornite in chiarezza, ed i certificati (in questo la struttura dell'esame ribadisce il ruolo metodologico ubiquito dei concetti di complessità computazionale propagandati nel corso). Tutto il resto (incluse le castronerie colossali ma anche le doppie risposte discordanti) non verrà conteggiato. Ricordate che in buona sostanza il voto corrisponderà al punteggio positivamente raccolto. I punti messi in palio ad ogni tema eccedono significativamente quanto necessario al raggiungimento dei pieni voti, gestitevi quindi per dimostrare le competenze che avete, senza impelagarvi dove avete invece delle lacune. Non ci interessano le vostre mancanze o lacune quanto piuttosto quello che dimostrate di saper fare.

L'esame presenta diverse tipologie di esercizi e domande su vari aspetti di quanto esposto a lezione. Nel prepararti all'esame, prendi a riferimento i testi e le correzioni dei temi precedenti che trovi al sito del corso:

<http://profs.sci.univr.it/~rrizzi/classes/RO/index.html>

Ogni esercizio è anche un'opportunità di apprendimento e di allenamento, sfruttalo al meglio senza sprecarlo. Una prima utilità è quella di testare la tua preparazione all'esame. Dopo aver letto il testo, consigliamo pertanto di svolgere l'esercizio quantomeno nella propria mente. Ma, in sufficiente numero di esemplari, poi anche materialmente, prestando attenzione ai tempi impiegati ed ai punti conseguiti. Solo a valle di un'esperienza almeno parziale con l'esercizio, passa alla lettura del documento di correzione. Se non sai come affrontare l'esercizio, sbircia sì la correzione, ma cercando di utilizzarla solo come suggerimento, cercando di riacquisire quanto prima autonomia nella conduzione dell'esercizio.

E se invece ti sembra di saper risolvere del tutto l'esercizio? Beh, a questo punto vale il converso: controlla che quanto hai in mente come soluzione corrisponda a quanto considerato e proposto come svolgimento opportuno. Nel confronto con la correzione proposta, presta attenzione non solo alle risposte in sè, ma anche a come esse vadano efficacemente offerte all'esaminatore/verificatore, ossia alla qualità dei tuoi certificati, alla precisione della tua dialettica, a come ottemperi il contratto implicito nella soluzione di un problema ben caratterizzato. In un certo senso, questo ti consentirà di raggiungere pragmaticamente quella qualità che in molti chiamano impropriamente ordine', che ha valore e giustamente finisce, volenti o nolenti, per essere riconosciuta in ogni esame della vita. Ordine, ma noi preferiamo chiamarlo saper rispondere in chiarezza alla consegna'' non significa bella calligrafia o descrizioni prolisse, ma cogliere tempestivamente gli elementi salienti, quelli richiesti da contratto più o meno implicito. In questo le competenze che abbiamo messo al centro di questo insegnamento di ricerca operativa potranno renderti più consapevolmente ordinato. Lo scopo del documento di correzione non è tanto quello di spiegare come l'esercizio vada risolto ma piuttosto come le risposte vadano adeguatamente esibite pena il mancato conseguimento dei punti ad esse associati. Aggiungo che per le tipologie di esercizio classiche, descrizioni curate dei più noti algoritmi risolutori possono essere facilmente reperite altrove (perchè non collaborare a raccogliere una ricca collezione di link a tali sorgenti?).