Esame di Ricerca Operativa - 28 settembre 2016
Facoltà di Scienze MM.FF.NN. - Verona
- CORREZIONE - punti in palio: 56, con voto $\geq$ punti

Problema 1 (5 punti):

La Skrinch-Skronch è un'azienda che produce snack. La disponibilità di materie prime, alla fine di gennaio, è la seguente: 550 kg di arachidi, 150 kg di pistacchi, 90 kg di mandorle e 70 kg di nocciole. Ogni scatola contiene 500 grammi di prodotto. La Skrinch-Skronch produce quattro tipi di snack, descritti di seguito:

prodotto	composizione	profitto (lire/scatola)
Mem	solo arachidi	260
Num	non piú del 50% di arachidi almeno il 10% di mandorle almeno il 15% di pistacchi	400
Pe	solo pistacchi	510
Qof	almeno il 30% di pistacchi almeno il 20% di mandorle almeno il 30% di nocciole	520

Supponendo che tutto quanto prodotto venga venduto, formulare come PL il problema di massimizzare il profitto della Skrinch-Skronch.**(4pt)** Indicare poi dove l'eventuale aggiunta di qualche vincolo di interezza possa lievemente aumentare la precisione del modello.**(1pt)**

svolgimento.

Il problema può essere formulato introducendo le seguenti variabili:

- x_{AM} = quantità di arachidi (in kg) utilizzate per produrre snack di tipo Mem;
- x_{AN} = quantità di arachidi (in kg) utilizzate per produrre snack di tipo Num;
- x_{MN} = quantità di mandorle (in kg) utilizzate per produrre snack di tipo Num;
- x_{NN} = quantità di nocciole (in kg) utilizzate per produrre snack di tipo Num;
- x_{PN} = quantità di pistacchi (in kg) utilizzati per produrre snack di tipo Num;
- x_{PP} = quantità di pistacchi (in kg) utilizzati per produrre snack di tipo Pe;
- x_{AQ} = quantità di arachidi (in kg) utilizzate per produrre snack di tipo Qof;
- x_{MQ} = quantità di mandorle (in kg) utilizzate per produrre snack di tipo Qof;
- x_{NQ} = quantità di nocciole (in kg) utilizzate per produrre snack di tipo Qof;
- x_{PQ} = quantità di pistacchi (in kg) utilizzati per produrre snack di tipo Qof;
- y_M = numero di scatole di snack di tipo Mem prodotte;
- y_N = numero di scatole di snack di tipo Num prodotte;
- y_P = numero di scatole di snack di tipo Pe prodotte;
- y_Q = numero di scatole di snack di tipo Qof prodotte.

Stiamo supponendo per semplicità che le variabili y_i non siano vincolate ad essere intere. L'obiettivo é quello di massimizzare i ricavi sulla vendita dei quattro tipi di confezioni ossia

$$\max R \;=\; 260\, y_M + 400\, y_N + 510\, y_P + 520\, y_Q\,,$$

nel rispetto dei seguenti vincoli:

vincoli di non negativitá

$$y_M, y_N, y_A, y_B, x_{AM}, x_{AN}, x_{MN}, x_{NN}, x_{PN}, x_{PP}, x_{AQ}, x_{MQ}, x_{NQ}, x_{PQ} \geq 0.$$

vincoli sulla composizione

$$\begin{aligned}
x_{AM} &= 0,5\, y_M \\
x_{AN} + x_{MN} + x_{PN} + x_{NN} &= 0,5\, y_N \\
x_{PP} &= 0,5\, y_P \\
x_{AQ} + x_{MQ} + x_{NQ} + x_{PQ} &= 0,5\, y_Q \\
x_{AN} &\leq 0,25\, y_N \\
x_{MN} &\geq 0,05\, y_N \\
x_{PN} &\geq 0,075\, y_N \\
x_{MQ} &\geq 0,1\, y_Q \\
x_{NQ} &\geq 0,15\, y_Q \\
x_{PQ} &\geq 0,15\, y_Q
\end{aligned}$$

disponibilitá di materie prime

$$\begin{aligned}
x_{AM} + x_{AN} + x_{AQ} &\leq 550 \\
x_{PP} + x_{PN} + x_{PQ} &\leq 150 \\
x_{MN} + x_{MQ} &\leq 90 \\
x_{NN} + x_{NQ} &\leq 70
\end{aligned}$$

Ovviamente i vincoli di non negativitá $y_M, y_N, y_A, y_B \geq 0$ possono essere omessi. Introducendo il vincolo di interezza per le sole 4 variabili y_M, y_N, y_A e y_B otteniamo soluzioni intere ottime che possono essere messe in pratica senza arrotondamenti (con conseguente rischio di perdita di precisione nella soluzione del modello matematico intero).

Problema 2 (3+2+1+3+8=17 punti):
Il professor Mod ha esteso la PL consentendo il riferimento a valori assoluti delle variabili nei termini della funzione obiettivo. Ha chiamato PL-mod il suo nuovo linguaggio di

Programmazione Matematica, e, nel suo libro, ha proposto l'esercizio di esprimere come un problema di PL il seguente problema di PL-mod.

$$
\min \; x_1 + 7|x_2| - 6x_3 + 9|x_4|
$$
$$
\begin{cases}
\begin{array}{cccccc}
x_1 + & x_2 + & x_3 + & x_4 & \leq & 18 \\
 & & & x_4 & \leq & 5 \\
 & & x_3 & & = & 7 \\
x_1 & + & x_3 & & \geq & 10
\end{array} \\
x_1, x_2, x_3, x_4 \in [-1000, 1000]
\end{cases}
$$

(3pt) si risolva l'esercizio proposto dal professor Mod.

(2pt) il seguente problema di PL-Mod viene chiamato il modello generale di Mod.

$$
\min \; \sum_{j=1}^{n} c_j x_j + \sum_{j=1}^{n} d_j x_j
$$
$$
\begin{cases}
\begin{array}{lcll}
\sum_{j=1}^{n} a_{i,j} x_j & \leq & b_i & i = 1, \ldots, r \\
\sum_{j=1}^{n} a_{i,j} x_j & \geq & b_i & i = r+1, \ldots, r+s \\
\sum_{j=1}^{n} a_{i,j} x_j & = & b_i & i = r+s+1, \ldots, r+s+t \\
x_j \in [L_j, U_j] & & & j = 1, \ldots, n
\end{array}
\end{cases} l
$$

Si parla di modello semi-generale di Mod nel caso particolare in cui $d_j \geq 0$ per ogni $j = 1, \ldots, n$. Si esprima attraverso la PL il modello semi-generale di PL-mod.

(1pt) Si dia un esempio di un problema, entro il modello semi-generale di Mod, dove la soluzione ottima sia unica ma non sia un vertice, ossia possa essere scritta come combinazione convessa di altre soluzioni ammissibili.

(3pt) Si esprima attraverso la PLI il modello generale di PL-mod.

(8=4+1+1+1+1pt) Si osservi come il ricorso alla PLI, senza limitarsi alla PL, fosse una rinuncia necessaria nell'ultimo dei casi visti sopra. Dimostrare questo osservando che l'ultimo e più generale dei modelli considerati è talmente espressivo da risultare NP-hard. Per fare ciò, suggeriamo di ridurre ad esso il problema PRECISELY-ONE-IN-3-SAT. In questo problema, noto essere NP-completo, viene data in input una formula di 3-SAT, ossia una formula booleana in CNF dove ogni clausola contiene 3 letterali, tutti e 3 positivi (ossia il NOT non compare mai nella formula), e si richiede di determinare se esista un assegnamento di verità alle variabili tale che, in ciascuna delle clausole, precisamente uno dei 3 letterali (cioè una variabile) valuti a *true*.

La sola descrizione di una riduzione valida vale 4 punti, mentre gli eventuali enunciati del lemma facile e difficile, e le loro eventuali dimostrazioni in breve, valgono 1 punto a testa.

svolgimento.

(3pt) A fianco delle variabili x_1, x_2, x_3, x_4 si introducano le variabili m_2 ed m_4 per le quali avremo modo di forzare $m_2 \geq |x_2|$ e $m_4 \geq |x_4|$. Il seguente problema di PL fornisce una riscrittura fedele del problema di PL-mod proposto dal Professor Mod.

$$
\min \; x_1 + 7m_2 - 6x_3 + 9m_4
$$
$$
\begin{cases}
m_2 \geq x_2 \\
m_2 \geq -x_2 \\
m_4 \geq x_4 \\
m_4 \geq -x_4 \\
x_1 + x_2 + x_3 + x_4 \leq 18 \\
x_4 \leq 5 \\
x_3 = 7 \\
x_1 + x_3 \geq 10 \\
x_1, x_2, x_3, x_4 \in [-1000, 1000]
\end{cases}
$$

(2pt) A fianco delle variabili x_j, $j = 1, \dots, n$, si introducano le variabili m_j, $j = 1, \dots, n$, per le quali avremo modo di forzare $m_j \geq |x_j|$. Il seguente problema di PL fornisce una riscrittura fedele del modello semi-generale di PL-mod. Stiamo quì assumendo che $d_j \geq 0$ per ogni $j = 1, \dots, n$.

$$
\min \; \sum_{j=1}^{n} c_j x_j + \sum_{j=1}^{n} d_j m_j
$$
$$
\begin{cases}
m_j \geq x_j & j = 1, \dots, n \\
m_j \geq -x_j & j = 1, \dots, n \\
\sum_{j=1}^{n} a_{i,j} x_j \leq b_i & i = 1, \dots, r \\
\sum_{j=1}^{n} a_{i,j} x_j \geq b_i & i = r+1, \dots, r+s \\
\sum_{j=1}^{n} a_{i,j} x_j = b_i & i = r+s+1, \dots, r+s+t \\
x_j \in [L_j, U_j] \;\; j = 1, \dots, n
\end{cases}
$$

(1pt) La soluzione $x = 0$ è ottima per il seguente problema di PL-Mod:

$$
\min \; -|x| \\
\{x \in [-1, 1]
$$

Si noti che $0 = \frac{1}{2}(-1) + \frac{1}{2}(1)$, ossia la soluzione ottima $x = 0$ è combinazione convessa delle soluzioni di base (vertici) $x = -1$ e $x = 1$, nessuna delle quali è però ottima. Sulla base di questa osservazione, la PL-Mod semi-generica appare come un arricchimento interessante della PL, i cui ottimi sono invece costretti a presentarsi sempre sui vertici.

(3pt) A fianco delle variabili x_j, $j = 1, \dots, n$, si introducano le variabili m_j, $j = 1, \dots, n$, per le quali avremo modo di forzare non solo $m_j \geq |x_j|$, ma di fatto $m_j = |x_j|$ sfruttando anche delle variabili booleane p_j ed n_j, con $p_j + n_j = 1$ e stanti ad indicare se il valore della x_j è **p**ositivo o **n**ositivo. Il seguente problema di PL fornisce una riscrittura fedele del modello generale di PL-mod.

$$
\min \ \sum_{j=1}^n c_j x_j + \sum_{j=1}^n d_j m_j
$$
$$
\begin{cases}
\sum_{j=1}^n a_{i,j}x_j & \leq b_i & i = 1,\ldots,r \\
\sum_{j=1}^n a_{i,j}x_j & \geq b_i & i = r+1,\ldots,r+s \\
\sum_{j=1}^n a_{i,j}x_j & = b_i & i = r+s+1,\ldots,r+s+t \\
x_j \geq & L_j & j = 1,\ldots,n \\
x_j \leq & U_j & j = 1,\ldots,n \\
m_j \geq & x_j & j = 1,\ldots,n \\
m_j \geq & -x_j & j = 1,\ldots,n \\
p_j, n_j \in & \{0,1\} & j = 1,\ldots,n \\
p_j + n_j = & 1 & j = 1,\ldots,n \\
m_j \leq & x_j + (U_j - L_j)n_j & j = 1,\ldots,n \\
m_j \leq & -x_j + (U_j - L_j)p_j & j = 1,\ldots,n
\end{cases}
$$

(4pt) Si assuma data una formula di PRECISELY-ONE-IN-3-SAT nelle variabili booleane $v_1,\ldots,v_p$ e nelle 3-clausole $T_1,\ldots,T_q$. Nel caso di PRECISELY-ONE-IN-3-SAT, una 3-clausola è la congiunzione di 3 letterali positivi (ossia senza che si applichi mai il NOT) sulle variabili $v_1,\ldots,v_p$, come ad esempio $T_t = v_i \vee v_j \vee v_k$. Sappiamo che decidere se esista un assegnamento di verità alle p variabili booleane tale che ogni clausola abbia precisamente una varibile posta a *true*), è un problema NP-completo. Per dimostrare l'NP-hardness del modello generale di Mod, vogliamo rappresentare entro esso la generica istanza di PRECISELY-ONE-IN-3-SAT data.

Per fare ciò, introduciamo le p variabili reali $x_1,\ldots,x_p$, che confineremo tutte nell'intervallo $[-1,1]$. L'idea è di utilizzare x_j per rappresentare v_j, per ogni $j=1,\ldots,n$, e secondo il seguente schema: $x_j = -1$ per rappresentare $v_j = \mathit{false}$ e $x_j = 1$ per rappresentare $v_j = \mathit{true}$, dove la spinta a far si che ogni variabile x_j vada a mapparsi sugli estremi dell'intervallo $[-1,1]$ concessole verrà dalla funzione obiettivo:

$$
-\left(\max \sum_{j=1}^p |x_j|\right) = \min \sum_{j=1}^p -|x_j|.
$$

Abbiamo quindi già i nostri "componenti bistabili" atti a rappresentare scelte binarie, ci manca solo rappresentare le clausole.

Chiediti ora, come faresti a rappresentare la generica clausola $T_t = v_a \vee v_b \vee v_c$?

Risposta: $x_a + x_b + x_c = -1$.

La descrizione della riduzione è completa **(4pt)** e sembra proprio confermare l'intuizione del Professor Mod che la PL-Mod sia ben più ricca ed espressiva della PL, rilevando tuttavia al tempo stesso quanto sia ambizioso il suo progetto di mettere a punto un algoritmo polinomiale per la PL-Mod.

Gli statement dei due lemmi cardine che sostengono la riduzione assicurano un ulteriore punto ciascuno.

(1pt) Lemma facile: se la PRECISELY-ONE-IN-3-SAT formula di partenza ammette un assegnamento di verità che pone a *true* precisamente una delle 3 variabili di ciascuna clausola, allora l'istanza di PL-Mod costruita come descritto ammette una soluzione di valore $-p$ (di fatto una soluzione ottima).

(1pt) Lemma difficile: se l'istanza di PL-Mod costruita come descritto ammette una soluzione di valore $-p$, allora la PRECISELY-ONE-IN-3-SAT formula di partenza ammette un assegnamento di verità che pone a *true* precisamente una delle 3 variabili di ciascuna clausola.

Anche se in generale non vi sarà il modo più conveniente di raccogliere punti, esemplifico delle possibili brevi dimostrazioni dei due lemmi. Se la dimostrazione contiene delle idee, ci si becca il punto. Se per sbaglio è anche corretta, si va già a crescere. E se fosse scritta con eleganza, sarebbe quì fuori luogo porre limiti.

($\geq$1pt) *Proof del Lemma facile:* Sia $\Phi : \{v_1, \ldots, v_p\} \mapsto \{false, true\}$ un assegnamento di verità alle variabili come ipotizzato nello statement del lemma. Si consideri $x_i = -1$ se $v_i = \mathit{false}$ e $x_i = 1$ se $v_i = \mathit{true}$. Si noti che $v_i \in [-1, 1]$ per ogni $i = 1, \ldots, p$, inoltre, per ogni clausola $T_t = v_a \vee v_b \vee v_c$ il vincolo associato $x_a + x_b + x_c = -1$ risulta soddisfatto in virtù dell'assunzione che in ciascuna delle clausole precisamente un letterale valuti a *true*. La soluzione proposta è pertanto ammissibile e vale min $\sum_{j=1}^{p} -|x_j| = -p$. □

($\geq$1pt) *Proof del Lemma difficile:* Si consideri una qualsiasi soluzione ammissibile di valore $-p$ per l'istanza di PL-Mod da noi costruita partendo dall'istanza di PRECISELY-ONE-IN-3-SAT. Poichè $\sum_{j=1}^{p} -|x_j| = -p$, ossia $\sum_{j=1}^{p} |x_j| = p$, e considerato che $|x_j| \leq 1$ per ogni $j = 1, \ldots, p$ visto che $x_j \in [-1, 1]$, allora $|x_j| = 1$ per ogni $j = 1, \ldots, p$, ossia $x_j \in \{-1, 1\}$. Siamo ora in condizioni di poter definire il seguente assegnamento di verità:

per ogni $j = 1, \ldots, p$, si prenda $v_j = \mathit{true}$ se $x_j = 1$ e $v_j = \mathit{false}$ se $x_j = -1$.

È facile verificare che tale assegnamento di verità necessariamente funziona: nella generica clausola $T_t = v_a \vee v_b \vee v_c$ precisamente uno dei letterali valuta a *true* in quanto il vincolo associato $x_a + x_b + x_c = -1$ risulta soddisfatto. □

Problema 3 (7 punti):
Un robot R, inizialmente situato nella cella A–1, deve portarsi nella sua home H situata nella cella G–9.

	1	2	3	4	5	6	7	8	9
A	*R*	1	3	1	1	1	0	0	•
B	2	2	1	0	•	•	0	0	0
C	2	•	0	1	0	0	1	1	1
D	0	0	1	0	0	0	1	•	0
E	0	0	•	1	•	1	0	0	0
F	0	1	1	1	0	3	•	0	1
G	3	•	0	1	2	0	0	1	*H*

I movimenti base possibili sono il passo verso destra (ad esempio dalla cella A–3 alla cella A–4) ed il passo verso in basso (ad esempio dalla cella A–3 alla cella B–3). Tuttavia il robot non può visitare le celle occupate da un pacman (•). Quanti sono i percorsi possibili? Inoltre, in ogni cella non occupata da un pacman (•) é presente un premio il cui valore è riportato nella cella stessa. Potremmo quindi essere interessati al massimizzare la somma dei valori dei premi raccolti lungo il percorso.

3.1(1pt) Quanti sono i percorsi possibili se la partenza è in A–1?

3.2 (1pt) e se la partenza è in B–3?

3.3 (1pt) e se con partenza in A–1 il robot deve giungere in F–6?

3.4 (1pt) e se con partenza in A–1 ed arrivo in G–9 al robot viene richiesto di passare per la cella D–5?

3.5(1pt) Quale é il massimo valore in premi raccoglibili lungo una traversata da A–1 a G–9?

3.6(2pt) Quanti sono i percorsi possibili che assicurino di portare a case tale massimo valore?

svolgimento. La risposta alle prime due domande può essere reperita nella rispettiva cella della seguente tabella di programmazione dinamica, dove in ogni cella C, partendo da quelle in basso a destra, si é computato il numero di percorsi che vanno dalla cella C alla cella G–9.

	1	2	3	4	5	6	7	8	9
A	256	168	102	36	8	8	8	2	•
B	88	66	66	28	•	•	6	2	1
C	22	•	38	28	18	11	4	1	1
D	22	14	10	10	7	7	3	•	1
E	8	4	•	3	•	4	3	3	1
F	4	4	4	3	2	1	•	2	1
G	0	•	1	1	1	1	1	1	1

Per rispondere alle due seguenti domande compilo un'ulteriore tabella, dove in ogni cella C, partendo da quelle in alto a sinistra, si computa il numero di percorsi che vanno dalla cella A–1 alla cella C.

	1	2	3	4	5	6	7	8	9
A	1	1	1	1	1	1	1	1	•
B	1	2	3	4	•	•	1	2	2
C	1	•	3	7	7	7	8	10	12
D	1	1	4	11	18	25	33	•	12
E	1	2	•	11	•	25	58	58	70
F	1	3	3	14	14	39	•	58	128
G	1	•	3	17	31	70	70	128	256

Ritrovare il valore 256 ci conforta, forse non abbiamo introdotto errori di calcolo nel computo delle due tabelle. La risposta alla terza domanda è contenuta nella rispettiva cella di questa seconda tabella.
La quarta domanda richiede di combinare le informazioni provenienti dalle due tabelle: la risposta è ottenuta come prodotto dei due valori riportati nelle due tabelle entro la cella di passaggio obbligato per il robot.

Per rispondere alle ultime due domande compilo un'ulteriore tabella, dove in ogni cella C, partendo da quelle in alto a sinistra, si computa il minimo costo di un percorso che va dalla cella A–1 alla cella C. Computiamo e riportiamo inoltre in piccolo, per ogni cella C, il numero di tali percorsi di costo minimo.

	1	2	3	4	5	6	7	8	9
A	0_1	1_1	4_1	5_1	6_1	7_1	7_1	7_1	•
B	2_1	4_1	5_2	5_3	•	•	7_1	7_2	7_2
C	4_1	•	5_2	6_5	6_5	6_5	8_1	9_1	10_1
D	4_1	4_1	6_2	6_7	6_{12}	6_{17}	9_1	•	10_1
E	4_1	4_2	•	7_7	•	7_{17}	9_1	9_1	10_1
F	4_1	5_3	6_3	8_7	8_7	11_7	•	9_1	11_1
G	7_1	•	6_3	9_7	11_7	11_{14}	11_{14}	12_{14}	12_{14}

Leggendo i valori riportati nella cella G–9 scopriamo che il minimo costo di una traversata é di 7, e che esistono 36 diversi possibili percorsi per raccogliere questo valore.

Riportiamo quindi i risultati finali.

consegna	numero percorsi
A–1 → G–9	256
B–3 → G–9	66
A–1 → F–6	39
passaggio per D–5	126
massimo valore	12
numero di min-val paths	14

Per maggiori e precise informazioni sulla logica con cui siano state compilate le varie tabelle di programmazione dinamica rimandiamo al codice c++ che le ha prodotte. Esso è reso disponibile nella stessa cartella della presente correzione.

Problema 4 (9 punti):

Una stringa è detta palindroma quando non fa differenza se viene letta da destra verso sinistra piuttosto che non da sinistra verso destra. Esempi: ANNA, AMA. Si consideri il problema di dover ricercare la più lunga sottosequenza palindoma entro una stringa. Esempio: ANONA in <u>AN</u>AC<u>ON</u>D<u>A</u>. Per trovare la più lunga sottosequenza palindroma in una stringa $s = s_1 s_2 \cdots s_n$, si è definita la seguente famiglia di sottoproblemi:

> per ogni $i, j \in \{1, 2, \ldots, n\}$ con $i \leq j$, sia $P[i, j]$ = la massima lunghezza di una sottosequenza palindroma di $s_{i,j} = s_i s_{i+1} \cdots s_j$.

4.1.**(3pt)** si dia la ricorrenza sulla famiglia dei sottoproblemi $P[i, j]$.

4.2.**(2pt)** si dia la base atta al calcolo di detta ricorrenza.

4.3.**(2pt)** ci si organizzi per trovare una sottosequenza palindroma di massima lunghezza in:
ACXCAXCAXAXXACXXACXCXAXCXAXCX

4.4.**(2pt)** ci si arrangi ed ingegni per trovare una sottosequenza palindroma di massima lunghezza in:
QY1CX6WCP7YXE2CY4RJAXDK9YSXVGXYC3XTM8BXY5CXACXNYFXCX

svolgimento o suggerimenti.

4.1.**(3pt)** la ricorrenza chiave:

$$P[i,j] = \begin{cases} 2 + P[i+1, j-1] & \text{se } s_i = s_j \\ \max\{P[i+1,j], P[i,j-1]\} & \text{altrimenti} \end{cases}$$

4.2.**(2pt)** la base atta all'avvio del calcolo per questa famiglia di problemi e ricorrenza:

$$P[i,i] = 1 P[i,i+1] = \begin{cases} 2 & \text{se } s_i = s_{i+1} \\ 1 & \text{altrimenti} \end{cases}$$

Si noti come la forma della ricorsione si rispecchi nel secondo caso base. Una possibile opzione era pertanto quella di introdurre anche i problemi "sentinella" della forma $P[i][i-1] = 0$.

4.3.**(2pt)** ci si organizzi per trovare una sottosequenza palindroma di massima lunghezza in: ACXCAXCAXAXXACXXACXCXAXCXAXCX

Collocherei i caratteri su una riga di un foglio a quadretti, lasciando un quadretto vuoto tra ogni due caratteri. Poi prenderei tre pennarelli per i colori A, C, ed X. Con un pennarello alla volta, individuerei uno ad uno i caratteri di quel colore, e, partendo dal carattere traccerei una linea con un primo tratto verticale di 1, e poi farei entrambe le diagonali a 45^o (come fare una Y).

Poi prendi una matita e scrivi degli 1 in tutti gli n quadretti in cui si aprono le Y. Poi, per ogni cella che contiene l'incrocio di due linee, partendo dalle celle più in basso e via via salendo, se le due linee che si incrociano hanno lo stasso colore, riporta nella cella il valore della cella sotto incrementato di 1, altrimenti riporta il massimo tra i valori immediatamente sotto, a 45^o a sinistra oppure a destra.

4.4.**(2pt)** ci si arrangi ed ingegni per trovare una sottosequenza palindroma di massima lunghezza in:

QY1CX6WCP7YXE2CY4RJAXDK9YSXVGXYC3XTM8BXY5CXACXNYFXCX

Scrivere $\binom{n}{2} = O(n^2)$ valori a mano, per n così grande sarebbe farsi del male, specie durante un esame organizzato come questo. Si noti però che alcuni caratteri compaiono una volta sola e quindi in definitiva il prenderli non è un'opzione da considerarsi.

Conviene prima dare una passata di preprocessing per eliminare tutti i caratteri che compaiano una volta sola, e poi agire come sopra. La risposta, invece di 22, questa volta dovrebbe essere 21. Il carattere centrale si specchia in se stesso.

Problema 5 (3 punti):
Dimostrare che K_5 è un grafo non planare.

dimostrazione 1. Il K_5 è un grafo con $n = 5$ nodi e $m = \binom{5}{2} = 10$ archi. Qualora il K_5 ammettesse un planar embedding, esso dovremme avere $f = m - n + 2 = 7$ facce per la formula di Eulero. Ciascuna di queste facce dovrebbe essere contornata de almeno 3 archi, visto che il K_5 non ha nè archi paralleli nè loops. Vero che ogni arco può servire a contornare 2 facce (contorna le due facce che separa), ma resta il fatto che ne seguirebbe l'assurdo $20 = 2m \geq 3f = 21$.

dimostrazione 2. Chiamiamo a,b,c,d ed e i 5 nodi del K_5. Anche in questa dimostrazione assumiamo di avere un embedding del K_5 e cerchiamo una contraddizione. I tre archi del triangolo a, b, c formeranno una curva chiusa e semplice, con una regione esterna ed una interna ad essa. I nodi d ed $e$non possono cadere uno nella regione interna ed uno in quella esterna poichè l'arco ed dovrebbe allora scavalcare la linea chiusa a, b, c. Assumiamo senza perdita di generalità che d ed e cadano entrambi nella regione interna. Siccome d cade internamente, i tre archi da, db ed dc separano tale regione interna in tre regioni:

- quella contornata dalla linea chiusa d, a, b lascia esterno il nodo c;
- quella contornata dalla linea chiusa d, b, c lascia esterno il nodo a;
- quella contornata dalla linea chiusa d, c, a lascia esterno il nodo b.

Ora, in qualsiasi di queste tre sottoregioni decida di cadere il nodo e, come potrà essere poi tracciato senza incroci l'arco che lo congiunge col nodo situato fuori da essa?

dimostrazione 3. Nell'embedding di K_5 il ciclo Hamiltoniano sui suoi nodi a, b, c, d, e è naturalmente mappato su una curva chiusa e semplice composta dai 5 archi tra ogni due nodi ciclicamente consecutivi. Restano da rappresentare altri 5 archi, e, senza perita di generalità, possiamo dedurne che almeno 3 di essi saranno mappati nella regione interna alla curva a, b, c, d, e. Si noti che almeno 2 di questi 3 archi, A_1 ed A_2, non hanno alcun estremo in comune dato che gli archi del K_5 non ricompresi nel ciclo Hamiltoniano costituiscono un secondo ciclo Hamiltoniano del K_5. Pertanto, quando si cammina lungo la curva a, b, c, d, e succede che gli estremi di questi archi si alternano: senza perdita di generalità, stiamo parlando degli archi ac e bd. Ed è chiaramente impossibile essi siano tracciati entrambi nella regione interna alla curva a, b, c, d, e senza incrociarsi tra loro.

Problema 6 (15 punti):
Si consideri il grafo, con pesi sugli archi, riportato in figura.

6.1.**(1pt)** Dire, certificandolo, se il grafo è planare oppure no. In ogni caso, disegnare il grafo in modo da minimizzare il numero di incroci tra archi.

6.2.**(1pt)** Dire, certificandolo, se il grafo ottenuto da G sostituendo l'arco hx con un arco qx è planare oppure no.

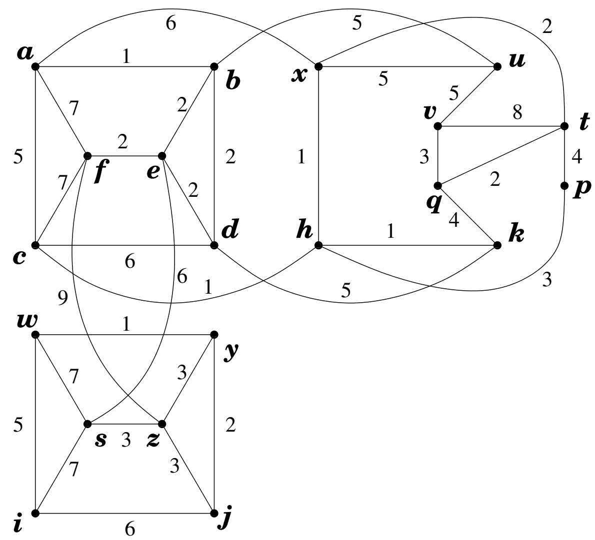
6.3.**(1+1pt)** Trovare l'albero dei cammini minimi dal nodo s. Esprimere la famiglia di tali alberi.

6.4.**(1pt)** Trovare un albero ricoprente di peso minimo.

6.5.**(4pt)** Per ciascuno dei seguenti archi dire, certificandolo, se esso appartenga a (tutte / a nessuna / a qualcuna ma non a tutte) le soluzioni ottime: dk, ax, es.

6.6.**(2pt)** Trovare tutti gli alberi ricoprenti di peso minimo. (Dire quanti sono e specificare con precisione come generarli).

6.7.**(1pt)** Trovare un massimo flusso dal nodo s al nodo t.

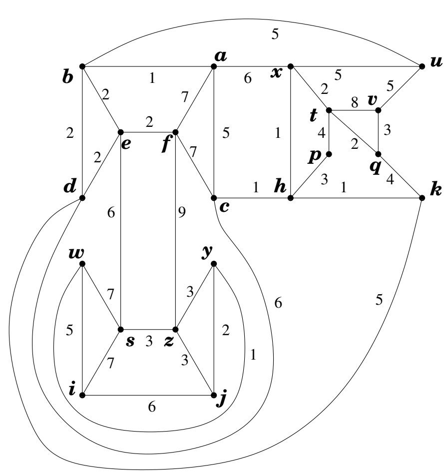


6.8.**(1pt)** Certificare l'ottimalità del flusso massimo dal nodo s al nodo t.

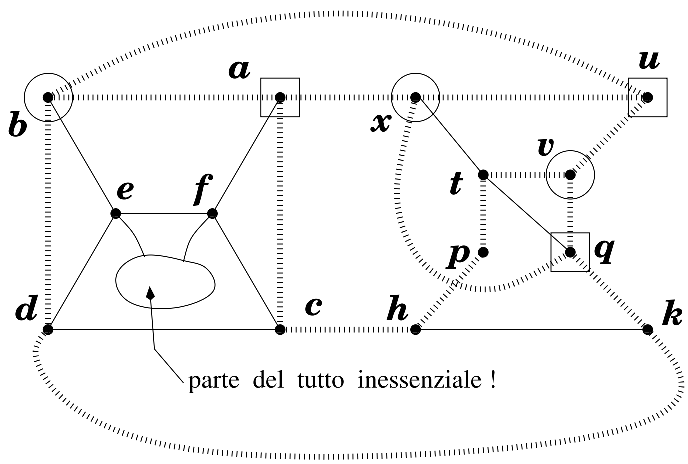
6.9.**(1+1pt)** Fornire (con certificato di ottimalità) il flusso massimo dal nodo s al nodo q.

risposte.

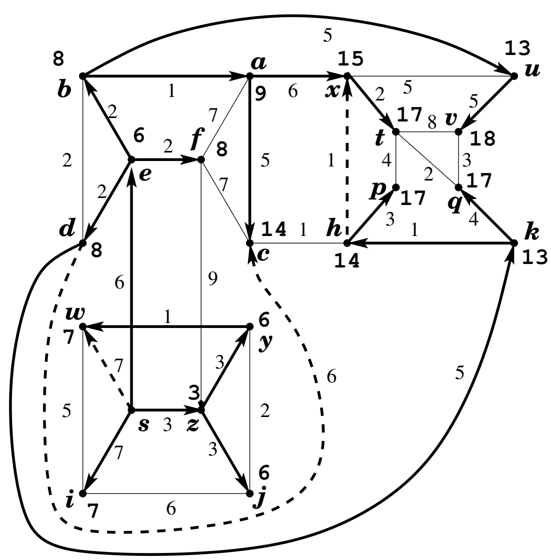
(1pt) Il fatto che G sia planare è messo in evidenza dal planar embedding fornito in figura.



Per la ricerca di alberi ricoprenti di peso minimo e di flussi massimi converrà lavorare sul planar embedding. E forse anche per osservare che il grafo modificato non è planare. Il certificato è la suddivisione del $K_{3,3}$ esibita in figura. **(1pt)**

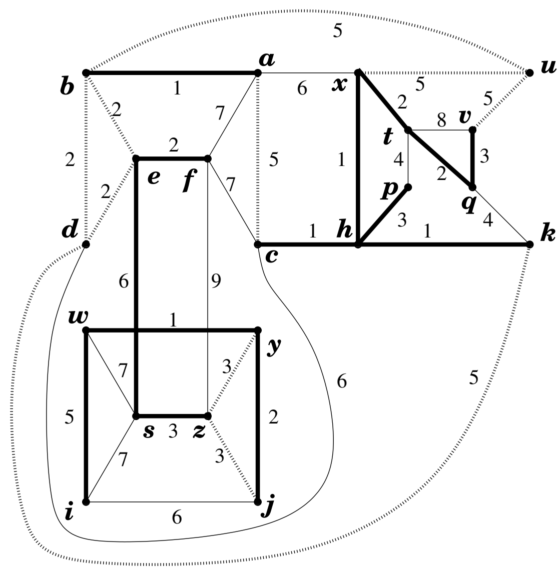


(1pt) Un albero dei cammini minimi dal nodo s a tutti gli altri nodi del grafo è rappresentato in figura dagli archi in linea spessa (sia tratteggiata che continua).

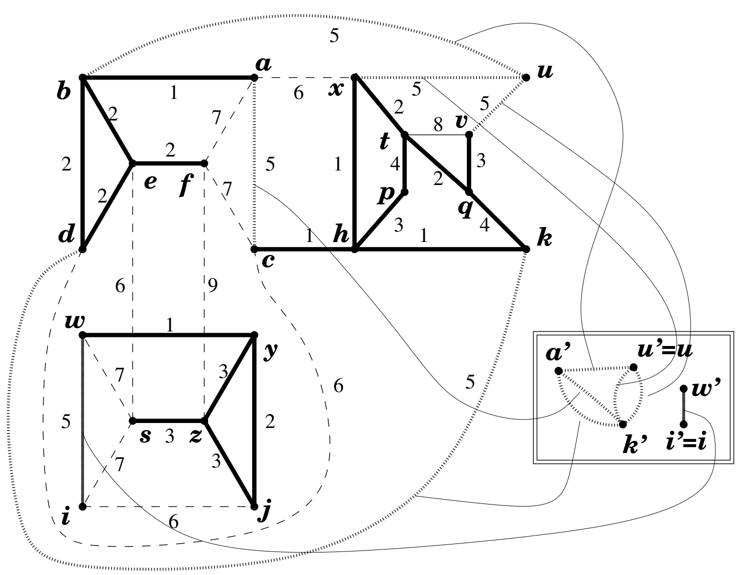


(1pt) Ovviamente ogni arco del grafo non contenuto nell'albero dei cammini minimi (ossia ogni arco in linea non spessa) può essere rimosso senza allontanare alcun nodo dal nodo s. Inoltre, anche i tre archi in linea spessa ma tratteggiata possono essere rimossi poichè sostituibili con altri archi (sempre in linea tratteggiata). Ci sono quindi $2^3 = 8$ alberi di cammini minimi dal nodo s: sono ottenuti aggiungendo all'albero fornito sopra un qualsiasi sottoinsieme dell'insieme di archi in linea tratteggiata e rimuovendo quegli archi in linea spessa che si trovino ad entrare in nodi dove ora entra un arco in linea tratteggiata.

(2+1pt) La seguente figura esprime la famiglia degli alberi ricoprenti di peso minimo.



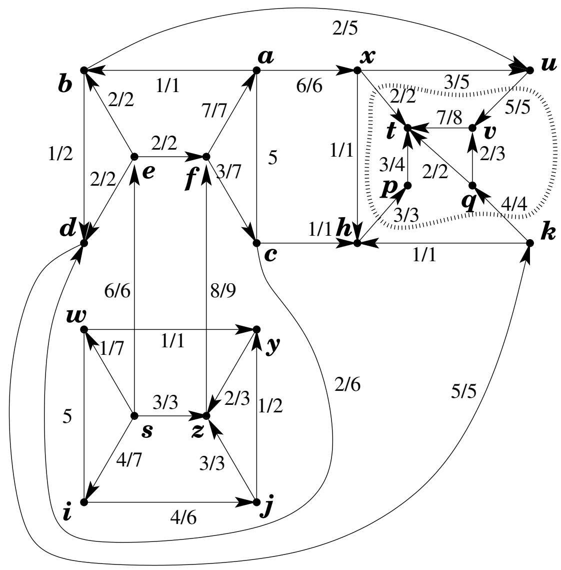
Ci sono $2 \cdot 3 \cdot 8 = 48$ alberi ricoprenti di perso minimo e ciascuno di essi include i 14 archi in linea spessa e nera, più 2 qualsiasi dei 3 archi di peso 2 in linea spessa e sfumata (3 scelte), più 1 qualsiasi dei 2 archi di peso 3 in linea spessa e sfumata (2 scelte), più 2 dei 5 archi di peso 5 ed in linea sfumata spessa, come da 8 possibili scelte che possono essere meglio comprese cancellando tutti gli archi di peso maggiore di 5 e contraendo tutti gli archi di peso inferiore a 5 come illustrato nella seguente figura.



(1pt) Un albero di peso minimo è il seguente: contiene tutti gli archi in linea nera grossa, più gli archi be e de, l'arco zy, e gli archi bu e vu.

(4pt) L'arco dk, appartiene a qualche soluzione ottima poichè di peso minimo (5) nel taglio costituito dagli archi: ax, ac, fc, dc e dk **(1pt)**; non appartiene tuttavia a tutte le soluzioni ottime poicheè arco di peso massimo (5) nel ciclo costituito dagli archi: db, bu, uv, vq, qk, kd **(1pt)**. L'arco ax, non appartiene ad alcuna soluzione ottima poichè unico arco di peso massimo (6) nel ciclo costituito dagli archi: ax, xh, hc, ca **(1pt)**. L'arco es, appartiene a tutte le soluzioni ottime poichè unico arco di peso minimo (5) nel taglio costituito dagli archi: es, fz **(1pt)**.

(1pt) La seguente figura esibisce un flusso massimo (non esibisco tutti i passaggi che ho dovuto compiere per ottenerlo) ed un taglio (minimo) che ne dimostra l'ottimalità.



Il flusso ha valore 14 e satura l'insieme degli archi (due soli archi) che attraversano la curva sfumata portandosi dal lato di s al lato di t. Questi 2 archi costituiscono pertanto un minimo s, t-taglio, anch'esso di valore 14 e che certifica pertanto l'ottimalità del flusso proposto. **(1pt)** (Sia il flusso che il taglio sarebbero stati più immediati a vedersi e verificarsi nel planar embedding. Puoi provare a rappresentarteli lì).

(1pt) Il massimo flusso da s a q a valore 9 e la stella di q è un taglio che ne certifica l'ottimalitá. **(1pt)** A parte questa strozzatura, vi è altrimenti ampio margine nell'inviare flusso da s a q ed evitiamo quindi di fornire descrizione di una tale soluzione ammissibile di valore 9 (cosa che ovviamente voi non potete mai fare: se volete totalizzare i rispettivi punti dovete innanzitutto fronirmi le soluzioni/certificati!). Essa può comunque essere facilmente prodotta con ovvie modifiche partendo dal flusso di valore 14 dato in figura.

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.
L'esame è scritto, dura circa 4 ore ed ha luogo in aula delta, dove, specie in estate, l'ambiente può risultare freddo. Consiglio di portarsi golfini, snack, acqua e matite o pennarelli colorati. (E dovete portare il tesserino col vostro numero di matricola.) Chi avesse problemi con l'aria condizionata è pregato di segnalarlo.
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 come scaricabili 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, usa pertanto il tuo senso critico per farne miglior uso senza sprecarlo. Una volta letto il testo di un esercizio, ti conviene sfruttarlo innanzitutto per testare la tua preparazione all'esame. Consigliamo pertanto di svolgere l'esercizio quantomeno nella propria mente, e comunque, su una buona percentuale di casi, anche materialmente (e prestando attenzione ai tempi impiegati ed ai punti conseguiti). Solo a valle di un'esperienza almeno parziale con l'esercizio, passa alla lettura della correzione. Se non sai come affrontare l'esercizio, sbircia sí la correzione, ma cercando di utilizzarla solo come suggerimento, cercando di riacquisire quanto prima autonomia nella conduzione dell'esercizio.

E una volta completato l'esercizio? Beh, a questo punto vale il converso: anche se ti sembra di avere svolto pienamente l'esercizio, omettere la successiva lettura della correzione, se fatto sistematicamente, rischia di rivelarsi una grave ingenuità. Il workflow standard cui riferirsi *cum granu salis* dovrebbe essere il seguente: esegui autonomamente l'esercizio e confronta poi le tue risposte con quelle nel rispettivo documento di correzione. 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 non conseguimento dei punti ad esse associati. È secondo quest'ottica che i documenti con le correzioni sono stati scritti. Preso cura di questo delicato aspetto (chiarire cosa si voglia dallo studente), altri obiettivi che, subordinatamente, cerco di assecondare nella stesura dei documenti di correzione sono semmai: aggiungere domande che arricchiscano l'esperienza di apprendimento offerta dall'esercizio, compendiare con altre considerazioni a latere che non potevano essere richieste allo studente, avanzare proposte di percorso ulteriore, e offrire spiegazioni contestualizzate che non possano essere reperite in altro documento. Infatti, per le tipologie di esercizio classiche, descrizioni curate dei più noti algoritmi risolutori possono essere facilmente reperite altrove (e vi incoraggio ad aiutarmi ad arricchire una tabella di link a tali sorgenti, o anche possiamo curare dispense di compendio a titolo di progetti che possono concorrere al voto).

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 mi interessano le vostre mancanze o lacune quanto piuttosto quello che dimostrate di saper fare. Se analizzate i temi di appelli precedenti, osserverete che avete a disposizione un'ampia varietà di modi per raccogliere punti e dimostrare la vostra preparazione. Lo scopo dell'esame sono il riconoscimento e la conferma. Essi sono a loro volta funzionali all'apprendimento. L'utilizzo corretto e pieno dei testi e correzioni rese disponibili ti consentirà di:

1. verificare la tua comprensione degli argomenti trattati e degli algoritmi e metodologie illustrati durante il corso;
2. affinare la tua preparazione ai fini dell'esame, non solo mettendo a punto le tue procedure ed approcci (privati e personali), ma chiarendo inoltre cosa l'esercizio richieda di produrre senza sbavature (ad esempio, a meno che non sia esplicitamente richiesto diversamente, la maggior parte degli esercizi non chiede che lo studente spieghi od illustri come ha risolto un problema, ma solo che fornisca risposte certificate);
3. toccare con mano la portata metodologica del concetto di certificato offertaci dalla complessità computazionale.

Durante l'esame, dovrete lavore 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 il 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, quelli che vi risultano più funzionali al concretizzare un buon punteggio. Il punteggio in buona sostanza corrisponde al voto. Contano le risposte corrette, fornite in chiarezza, ed i certificati. Tutto il resto non verrà conteggiato. In questo la struttura dell'esame ribadisce il ruolo metodologico ed ubiquito dei concetti di complessità computazionale propagandati nel corso.

gestione dei voti conseguiti.

I voti dei singoli appelli verrano comunicati e resi disponibili tramite ESSE3. Dal 18 in sù i voti verranno registrati automaticamente a valle di un intervallo di tempo concessovi per eventualmente rifiutare il voto.

L'eventuale rifiuto del voto, oppure la sua sospensione (per condurre un progetto atto ad incrementare il voto, oppure perchè lo studente richiede del tempo per pensarci, oppure chiede di poter partecipare ad appello successivo decidendo solo alla fine se consegnare o meno riscrivendo voto precedente) vanno richiesti con una mail. Ovviamente, specie per un progetto, se ne deve parlare anche a voce, ma la mail serve comunque come promemoria e contabilità.

Se hai idee su come migliorare il corso od i suoi materiali proponi un tuo progetto, con esso potrai aggiungere al voto conseguito all'esame.