17) Risolvere il seguente problema usando i tagli di Gomory:

$$\min \left\{ p' = -4x_1 - 3x_2 : \begin{array}{l} 2x_1 + x_2 \le 11 \\ -x_1 + 2x_2 \le 6 \\ x_1, x_2 \in \mathbb{N} \end{array} \right\}$$

17) TABLEAU:

NB: $z = p'$ Infatti:

$$\min \left\{ z = -4x_1 - 3x_2 : \begin{array}{l} 2x_1 + x_2 + x_3 = 11 \\ -x_1 + 2x_2 + x_4 = 6 \\ x_1, x_2, x_3, x_4 \ge 0 \end{array} \right\}$$

		x_1	x_2	x_3	x_4
$-z$	0	-4	-3	0	0
x_3	11	2	1	1	0
x_4	6	-1	2	0	1

I

$\Rightarrow$

		x_1	x_2	x_3	x_4
z	22	0	-1	2	0
x_1	$\frac{11}{2}$	1	$\frac{1}{2}$	$\frac{1}{2}$	0
x_4	$\frac{23}{2}$	0	$\frac{5}{2}$	$\frac{1}{2}$	1

$\Rightarrow$

		x_1	x_2	x_3	x_4
z	$26+\frac{3}{5}$	0	0	$\frac{11}{5}$	$\frac{2}{5}$
x_1	$\frac{16}{5}$	1	0	$\frac{2}{5}$	$-\frac{1}{5}$
x_2	$\frac{23}{5}$	0	1	$\frac{1}{5}$	$\frac{2}{5}$

A questo punto prendiamo la prima riga come riga generatrice.

$$x_1 + 0x_2 + \tfrac{2}{5}x_3 - \tfrac{1}{5}x_4 = \tfrac{16}{5} \Rightarrow x_1 + 0x_2 + \left(0 + \tfrac{2}{5}\right)x_3 + \left(-1 + \tfrac{4}{5}\right)x_4 = 0 + \tfrac{16}{5}$$

Quindi:

$$-\tfrac{16}{5} \ge -x_1 - \tfrac{2}{5}x_3 - \tfrac{4}{5}x_4$$

Dopo di che si ottiene il tableau orlato.

		x_1	x_2	x_3	x_4	x_5
$-z$	$26+\frac{3}{5}$	0	0	$\frac{11}{5}$	$\frac{2}{5}$	
x_1	$\frac{16}{5}$	1	0	$\frac{2}{5}$	$-\frac{1}{5}$	0
x_2	$\frac{23}{5}$	0	1	$\frac{1}{5}$	$\frac{2}{5}$	0
x_5	$-\frac{16}{5}$	0	0	$-\frac{2}{5}$	$-\frac{4}{5}$	1

Infatti: $-\frac{16}{5} = -\frac{2}{5}x_3 - \frac{4}{5}x_4 + x_5$

Si prosegue poi con il metodo.

18) Dato il problema:

$$\min \left\{ z = 3x_1 + 4x_2 + 2x_3 : \begin{array}{l} x_1 + 4x_2 + x_3 \ge 3 \\ 3x_1 - x_2 + 2x_3 \ge 2 \\ x_1, x_2, x_3 \ge 0 \end{array} \right\}$$

a) Scrivere il problema duale

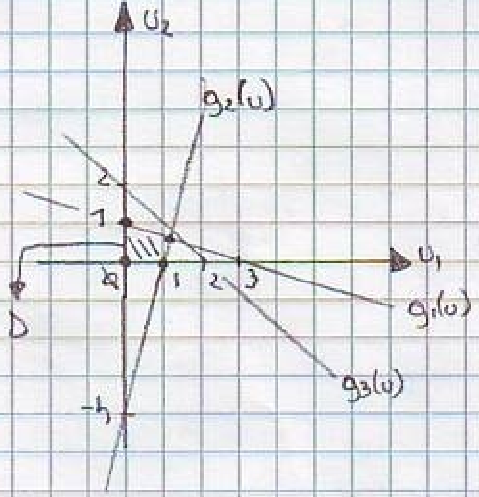
b) Dato il duale usare gli scarti complementari per risolvere il problema primale.

18) a) Problema duale:

$$\left\{ \max z = 3u_1 + 2u_2 : \begin{array}{l} u_1 + 3u_2 \le 3 \\ 4u_1 - u_2 \le 4 \\ u_1 + u_2 \le 2 \\ u_1, u_2 \ge 0 \end{array} \right\}$$

NB: $[u_1 \; u_2] \cdot \begin{bmatrix} 1 & 4 & 1 \\ 3 & -1 & 2 \end{bmatrix} = \begin{bmatrix} u_1 + 3u_2 \\ 4u_1 + (-u_2) \\ u_1 + u_2 \end{bmatrix}$

b) Risoluzione grafica:



$g_1(u): u_1 + 3u_2 \le 3$

$g_2(u): 4u_1 - u_2 \le 4$

$g_3(u): u_1 + u_2 \le 2$

Grad $z = 3, 2$

In particolare: $A = (0,0)$
$B = (0,1)$
$C = (1{,}2, 0{,}6)$
$D = (1,0)$

Infatti: $\begin{cases} u_1 + 3u_2 \le 3 \\ 4u_1 - u_2 \le 4 \end{cases}$

$\Downarrow$

$u_1 = 3 - 3u_2$

$4(3 - 3u_2) - u_2 = 4$

Quindi: $12 - 12u_2 - u_2 = 4$

$\Downarrow$

$8 = 13u_2 \Rightarrow \underline{u_2 = 8/13 \cong 0{,}6}$

Poi: $\underline{u_1 \cong 1{,}2}$

L'ottimo è il punto C.
Metodo degli scarti complementari:

$x_1^*(3 - u_1^* - 3u_2^*) = 0$

$x_2^*(4 - 4u_1^* + u_2^*) = 0$

$x_3^*(2 - u_1^* - 2u_2^*) = 0$

$(x_1^* + 4x_2^* + x_3^* - 3)\, u_1^* = 0$

$(3x_1^* - x_2^* + 2x_3^* - 2)\, u_2^* = 0$

$\Rightarrow x_1^*(3 - 1{,}2 - 3 \cdot 0{,}6) = 0 \Rightarrow \underline{x_1^* \neq 0}$

$x_2^*(4 - 4 \cdot 1{,}2 + 0{,}6) = 0 \Rightarrow x_2^* = 0$

$x_3^*(2 - 1{,}2 - 2 \cdot 0{,}6) = 0 \Rightarrow x_3^* = 0$

$\Downarrow$

... ma la soluzione non è corretta!!

Infatti: $u_1^* = 10/9$
$u_2^* = 4/9 \Rightarrow x_1^* = 0$
$4x_2^* + x_3^* = 3$
$3x_1^* - x_2^* = 2$
$\Rightarrow x_1^* = 0$
$x_2^* = 4/9$
$x_3^* = 11/9$

19) Un escursionista deve preparare lo zaino. Vuole portare 5 oggetti. Peso massimo $P_{max} = 35$.

v_j	30	11	15	105	112
p_j	5	3	3	15	16

Risolvere il problema con il branch & bound.

19)

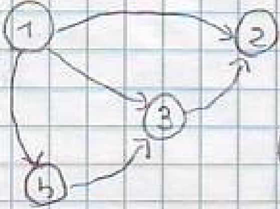
52

Problema: $\max\left\{ z = \sum_j v_j x_j : \sum_j p_j x_j \leq P, \; x_j \in \{0,1\} \right\}$

Questa seconda parte di ricerca operativa tratta i GRAFI. Un GRAFO è una "struttura dati" del tipo:

$G = \langle N, A \rangle$ dove $\begin{cases} N = \text{INSIEME DEI NODI} \\ A = \text{INSIEME DI ARCHI} \end{cases}$

Graficamente:

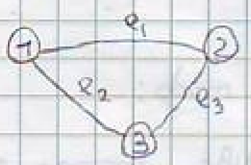


$\begin{cases} N = \{1,2,3,4\} \\ A = \{(1,2),(1,3),(1,4),(4,3),(3,2)\} \end{cases}$

Per essere più precisi abbiamo due tipi di grafi:

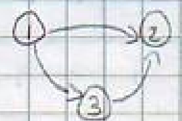
1) GRAFO NON ORIENTATO
2) GRAFO ORIENTATO (DIGRAFO)

Un grafo non orientato è un grafo dove gli archi non hanno un orientamento, e prendono quindi il nome di LATI. Per esempio:



V = INSIEME DEI VERTICI = $(1,2,3)$
E = INSIEME DEI LATI = (e_1, e_2, e_3).

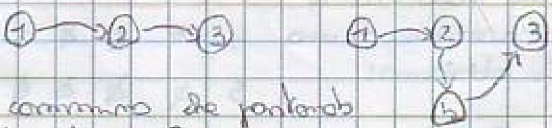
Un digrafo è invece una struttura di questo tipo:



$N = \{1,2,3\}$
$A = \{(1,2),(1,3),(3,2)\}$

Si chiama CAMMINO una sequenza di archi in un digrafo. Per esempio:

CAMMINO: 1, 2, 3 oppure 1, 2, 4, 3.

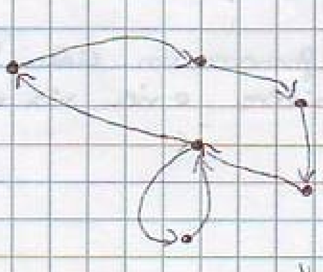


Un CICLO è un cammino chiuso, cioè un cammino che partendo da un dato nodo permette di tornare al nodo stesso. Per esempio:

Un ciclo può essere di due tipi:
1) EULERIANO
2) HAMILTONIANO

Un ciclo si dice euleriano quando tocca tutti i lati del grafo una e una sola volta. Un ciclo si dice hamiltoniano quando tocca tutti i vertici del grafo una e una sola volta. Per esempio:

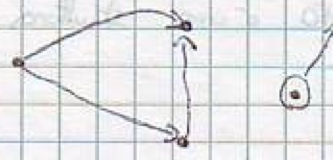
CICLO HAMILTONIANO



CICLO EULERIANO (ma non hamiltoniano.)

In particolare un ALBERO è un sottografo di un grafo che possiede tutti i vertici del grafo ed è connesso ma senza cicli. Un grafo connesso è un grafo che non ha nodi isolati. Per esempio

(54)



NODO ISOLATO

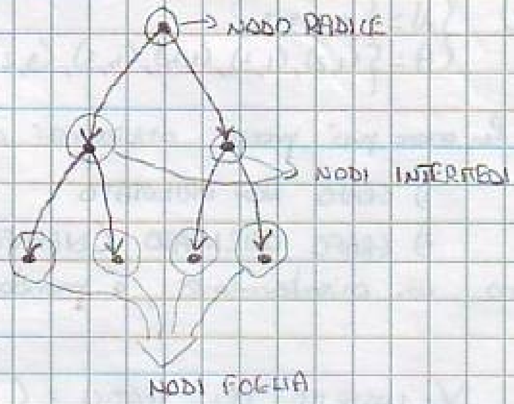
non è un grafo connesso.

Formalmente: $\forall V_i, V_j \in V$

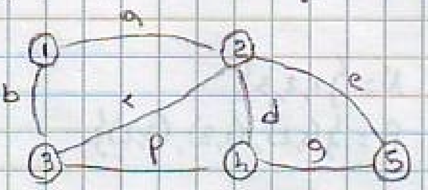
$\Downarrow$

$\exists V_i \to V_j$

Un esempio di albero è il seguente:



Un grafo può essere rappresentato matricialmente nel seguente modo:



	A	B	C	D	E	F	G
1	1	1	Ø	Ø	Ø	Ø	Ø
2	1	Ø	1	1	1	Ø	Ø
3	Ø	1	1	Ø	Ø	1	Ø
4	Ø	Ø	Ø	1	Ø	1	1
5	Ø	Ø	Ø	Ø	1	Ø	1

NB:  = ARCO SEMPLICE

= ARCO MULTIPLO

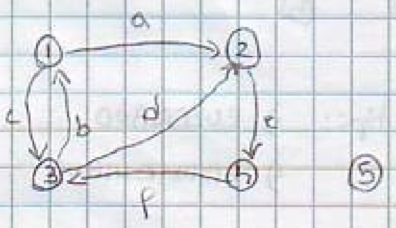
NB: 

GRADO USCENTE = 2

GRADO ENTRANTE = 2

GRADO = 3

"1" indica il fatto che l'arco in questione incide sul nodo. La "Ø" sta a indicare l'assenza di una tale incisione. Se abbiamo:



$\to$

	A	B	C	D	E	F
1	1	-1	1	Ø	Ø	Ø
2	-1	Ø	Ø	-1	1	Ø
3	Ø	1	-1	1	Ø	-1
4	Ø	Ø	Ø	Ø	-1	1

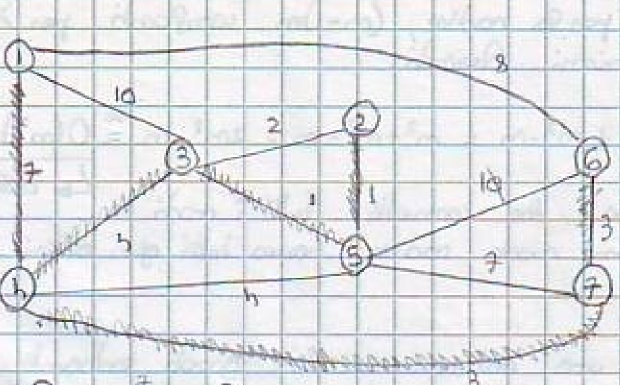
$\hookrightarrow$ MATRICE DI INCIDENZA.

NB:

Vediamo ora alcuni algoritmi legati ai grafi:

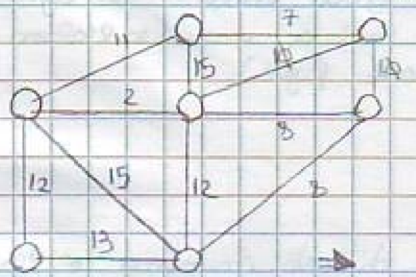
1) ALGORITMO DI KRUSKAL: Trovare albero T (ALBERO RICOPRENTE) di costo minimo che connette tutti i vertici del grafo $G = (V, E)$ dotato di una funzione costo: $c: E \to \mathbb{Z}$.
Non bisogna prendere i lati che formano un ciclo. Si prendono i lati partendo da quello di costo minore e via via verso quello di costo maggiore.

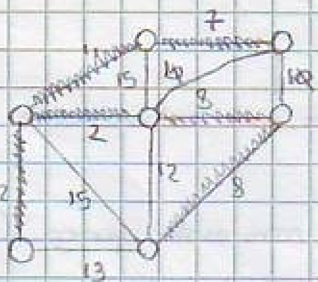
Vediamo un esempio:

1) 

COSTI IN ORDINE NON DECRESCENTE:

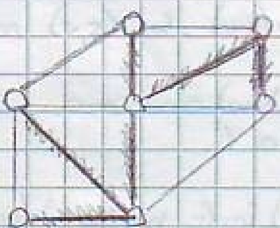
(3,5)
(2,5)
(2,3) no perché fa ciclo
(4,7)
(6,7)
(3,4)
(4,5) no perché fa ciclo
(5,7) " "
(1,4)
(1,6) no perché fa ciclo
(1,3) " "
(5,6) " "

2) 



ALBERO DI COSTO MINIMO

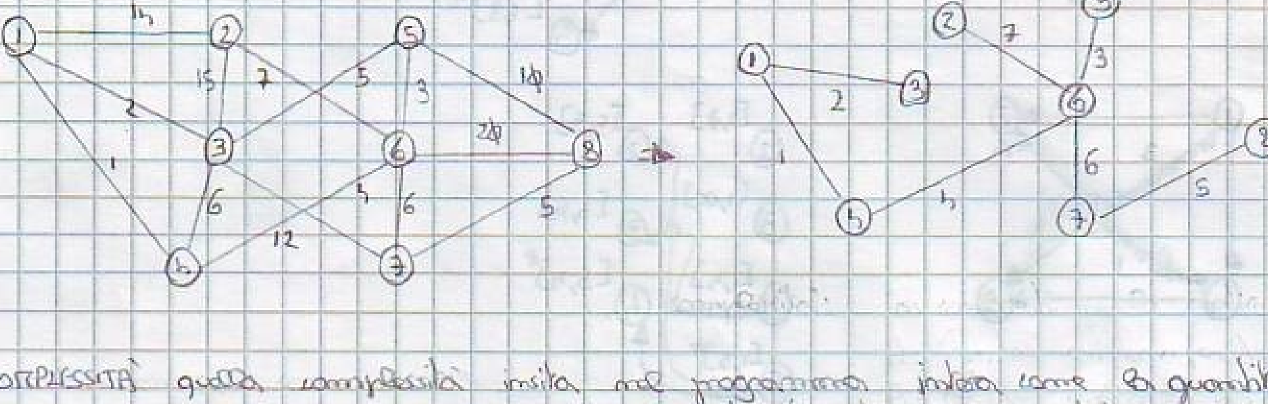
L'albero di costo massimo si ottiene:



Analizziamo ora l'algoritmo di PRIM. Esso funziona così:

- Scelgo $v \in V$ di G come radice dell'albero. Pongo $V' = \{v\}$, $\forall w \neq v \Rightarrow \lambda(w) = c(v,w)$, $E' = \emptyset$. Se (v,w) non esiste pongo $\lambda(w) = \infty$
- Trovo $w \in V/V'$ con $\lambda(w)$ minimo, e sia 'e' un lato che connette w a V' con costo $\lambda(w)$. Aggiungo 'e' a E' e w a V'.
- $\forall w' \notin V'$ pongo $\lambda(w') = \min(\lambda(w'), c(w,w'))$
- Se $V' = V$ STOP, altrimenti torna al passo 2.

Esempio:



Definiamo COMPLESSITÀ quella complessità insita nel programma intesa come la quantità di risorse necessarie per la sua esecuzione. Ci sono due tipi di complessità:

1) SPAZIALE
2) TEMPORALE.

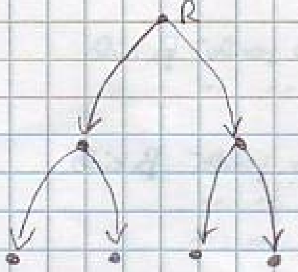
Noi considereremo la complessità temporale.

L'algoritmo di PRIM svolge n confronti per la radice, $(n-1)n$ confronti per $\lambda(w)$ minimo e $(n-1)n$ confronti per $c(w,w')$ e $\lambda(w)$ minimi. Quindi:

$$n + n(n-1) + n(n-1) = \cancel{n} + n^2 - \cancel{n} + n^2 - n = n^2 + n^2 - n = 2n^2 - n \cong O(n^2).$$

↳ dell'ordine di n^2.

Definiamo un'ARBORESCENZA come un albero che connette tutti i nodi di un grafo G, orientato a partire da un nodo radice verso tutti gli altri nodi o viceversa.

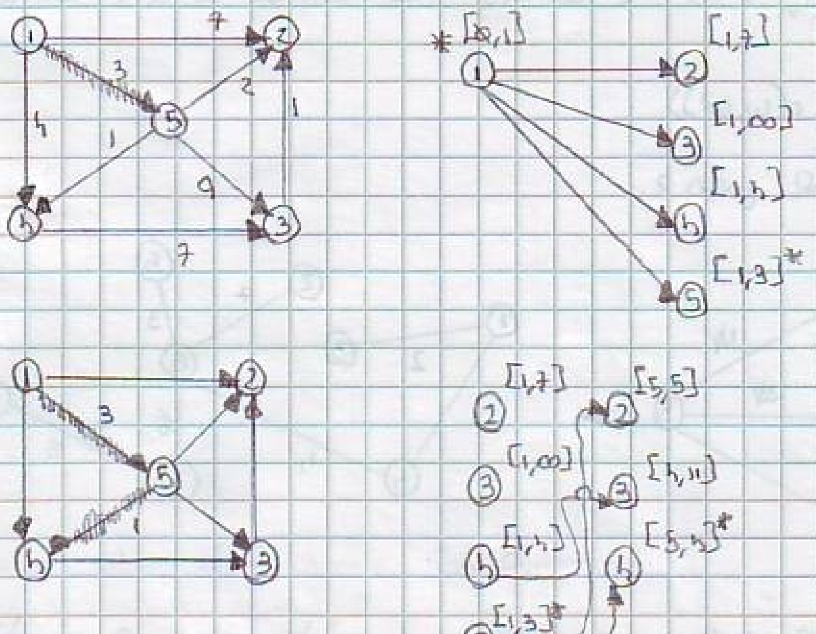


Proprietà: In una arborescenza il grado entrante in ogni nodo diverso dalla radice è pari a uno. (se l'albero è diretto dalla radice verso le foglie).

Un altro algoritmo che ci permette di trovare un'arborescenza di costo minimo è l'algoritmo di DIJKSTRA.

- $\forall v \in V/\{s\}$ pongo $\lambda(v) = c(s,v)$. Se (s,v) non esiste pongo $\lambda(v) = \infty$. Inoltre: $V' = \{s\}$, $A' = \Phi$
- Determino $w \in V/V'$ con $\lambda(w)$ minima. Sia $v \in V'$ tale che: $\lambda(w) = \lambda(v) + c(v,w)$. Aggiungo (v,w) ad A' e 'w' a V'.
- $\forall w' \notin V'$ per cui $\exists (w,w')$ in G pongo $\lambda(w') = \min(\lambda(w'), \lambda(w) + c(w,w'))$
- se $V' = V$ STOP, altrimenti torna al passo 2.

Si noti che V/V' indica l'insieme contenente elementi di V ma non di V'. Anche l'algoritmo di Dijkstra è dell'ordine di $O(n^2)$ in termini di complessità. In questo algoritmo, la funzione costo deve essere intera ma non negativa. Per esempio:



═ = arco preso in considerazione