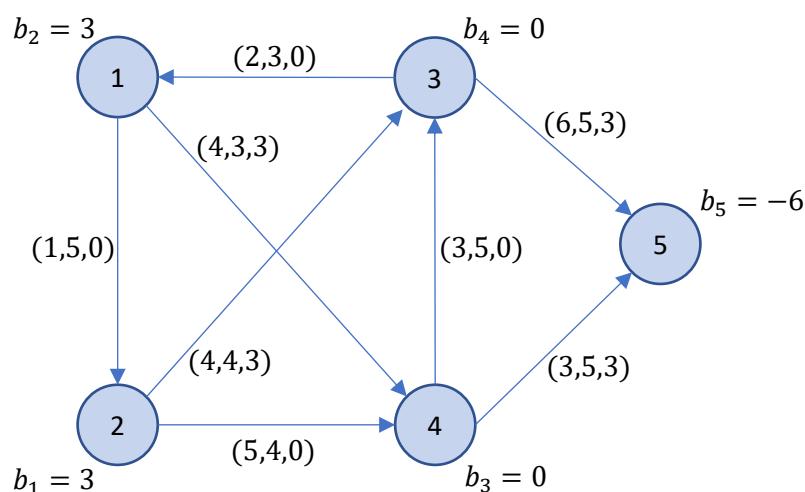


Teoria dei Grafi – Parti 1 e 2

Esercizi

Gli esercizi presentati qui di seguito hanno lo svolgimento completo, però all'esame potrebbe essere richiesto di svolgere solo uno o due iterazioni, partendo dall'inizio oppure da una soluzione intermedia fornita nel testo. Nello svolgimento sono stati inseriti dei commenti che suggeriscono le possibili varianti delle domande proposte nell'esame scritto.

- 1) Si consideri il seguente grafo G:



Su ogni arco (i,j) è riportata la tripletta (c_{ij}, u_{ij}, x_{ij}) , dove c_{ij} è il costo per trasportare una unità di flusso, u_{ij} è la capacità e x_{ij} è il flusso corrente (a seconda delle domande che seguono potranno essere usati anche solo alcuni dei dati forniti).

- a) Determinare il flusso massimo partendo dal flusso nullo (i.e., $x_{ij} = 0$, per ogni arco (i,j)).

Algoritmo Flusso Massimo

Grafo Iniziale Flusso Massimo

Nodo 1: $x(1,2)=0 \quad x(1,4)=0$

Nodo 2: $x(2,3)=0 \quad x(2,4)=0$

Nodo 3: $x(3,1)=0 \quad x(3,5)=0$

Nodo 4: $x(4,3)=0 \quad x(4,5)=0$

Etichetta nodo 1: [1,1000]

Etichetta nodo 2: [1,5]

Etichetta nodo 4: [1,3]

Etichetta nodo 3: [2,4]

Etichetta nodo 5: [3,4]

Aumenta il flusso di 4 nel cammino: (1,2) (2,3) (3,5)

Etichetta nodo 1: [1,1000]

Etichetta nodo 2: [1,1]

Etichetta nodo 4: [1,3]

Etichetta nodo 3: [4,1]

Etichetta nodo 5: [4,3]

Aumenta il flusso di 3 nel cammino: (1,4) (4,5)

Etichetta nodo 1: [1,1000]

Etichetta nodo 2: [1,1]

Etichetta nodo 4: [2,1]

Etichetta nodo 3: [4,1]

Etichetta nodo 5: [4,1]

Aumenta il flusso di 1 nel cammino: (1,2) (2,4) (4,5)

Etichetta nodo 1: [1,1000]

Flusso Massimo Trovato!

Numero Iterazioni: 4

Risultato Flusso Massimo

Nodo 1: $x(1,2)=5$ $x(1,4)=3$

Nodo 2: $x(2,3)=4$ $x(2,4)=1$

Nodo 3: $x(3,1)=0$ $x(3,5)=4$

Nodo 4: $x(4,3)=0$ $x(4,5)=4$

Flusso Massimo = 8

- b) Determinare il flusso di costo minimo con l'algoritmo Cycle Cancelling generando una soluzione ammissibile (e non utilizzando i flussi x_{ij} forniti nel testo). [Nota: in questo caso all'esame potremo chiedere di generare solo la soluzione ammissibile, oppure svolgere solo la prima iterazione oppure le prime due]

Algoritmo Flusso di Costo Minimo: senza soluzione ammissibile

Cerca Flusso Ammissibile

Etichetta nodo 6: [6,1000]

Etichetta nodo 1: [6,3]

Etichetta nodo 2: [6,3]

Etichetta nodo 4: [1,3]

Etichetta nodo 3: [2,3]

Etichetta nodo 5: [3,3]

Etichetta nodo 7: [5,3]

Aumenta il flusso di 3 nel cammino: (6,2) (2,3) (3,5) (5,7)

Etichetta nodo 6: [6,1000]

Etichetta nodo 1: [6,3]

Etichetta nodo 2: [1,3]

Etichetta nodo 4: [1,3]

Etichetta nodo 3: [2,1]

Etichetta nodo 5: [3,1]

Etichetta nodo 7: [5,1]

Aumenta il flusso di 1 nel cammino: (6,1) (1,2) (2,3) (3,5) (5,7)

Etichetta nodo 6: [6,1000]

Etichetta nodo 1: [6,2]

Etichetta nodo 2: [1,2]

Etichetta nodo 4: [1,2]

Etichetta nodo 3: [4,2]

Etichetta nodo 5: [4,2]

Etichetta nodo 7: [5,2]

Aumenta il flusso di 2 nel cammino: (6,1) (1,4) (4,5) (5,7)

Etichetta nodo 6: [6,1000]

Flusso Massimo Trovato!

Numero Iterazioni: 4

Soluzione Ammissibile Iniziale

Nodo 1: $x(1,2)=1$ $x(1,4)=2$

Nodo 2: $x(2,3)=4$ $x(2,4)=0$

Nodo 3: $x(3,1)=0$ $x(3,5)=4$

Nodo 4: $x(4,3)=0$ $x(4,5)=2$

Cerca Cicli di Costo Negativo

Iterazione: 1

Ciclo: $C(1,4)=4$ $C(4,5)=3$ $C(5,3)=-6$ $C(3,2)=-4$ $C(2,1)=-1$

Aumento flusso di: 1

Iterazione: 2

Ciclo: $C(2,4)=5$ $C(4,5)=3$ $C(5,3)=-6$ $C(3,2)=-4$

Aumento flusso di: 2

Iterazione: 3

Ciclo non trovato: Flusso Costo Minimo Trovato

Numero Iterazioni: 3

Risultato Flusso di Costo Minimo
Nodo 1: $x(1,2)=0$ $x(1,4)=3$
Nodo 2: $x(2,3)=1$ $x(2,4)=2$
Nodo 3: $x(3,1)=0$ $x(3,5)=1$
Nodo 4: $x(4,3)=0$ $x(4,5)=5$

Costo Soluzione = 47

- c) Determinare il flusso di costo minimo con l'algoritmo Cycle Cancelling partendo dai flussi x_{ij} forniti nel testo. [Nota: in questo caso all'esame potremo chiedere svolgere solo la prima iterazione oppure le prime due]

Algoritmo Flusso di Costo Minimo: con soluzione ammissibile

Soluzione Ammissibile Iniziale
Nodo 1: $x(1,2)=0$ $x(1,4)=3$
Nodo 2: $x(2,3)=3$ $x(2,4)=0$
Nodo 3: $x(3,1)=0$ $x(3,5)=3$
Nodo 4: $x(4,3)=0$ $x(4,5)=3$

Cerca Cicli di Costo Negativo

Iterazione: 1
Ciclo: $C(2,4)=5$ $C(4,5)=3$ $C(5,3)=-6$ $C(3,2)=-4$
Aumento flusso di: 2

Iterazione: 2
Ciclo non trovato: Flusso Costo Minimo Trovato

Numero Iterazioni: 2

Risultato Flusso di Costo Minimo
Nodo 1: $x(1,2)=0$ $x(1,4)=3$
Nodo 2: $x(2,3)=1$ $x(2,4)=2$
Nodo 3: $x(3,1)=0$ $x(3,5)=1$
Nodo 4: $x(4,3)=0$ $x(4,5)=5$

Costo Soluzione = 47

- d) Determinare il flusso di costo minimo con l'algoritmo Successive Shortest Path partendo dai flussi x_{ij} nulli e potenziali π_i nulli. [Nota: in questo caso all'esame potremo chiedere svolgere solo la prima iterazione oppure le prime due. In alternativa potremo fornire il flusso e i potenziali e chiedere di svolgere una o due iterazioni]

Algoritmo Flusso di Costo Minimo: Successive Shortest Path

Soluzione Iniziale
Nodo 1: $x(1,2)=0$ $x(1,4)=0$
Nodo 2: $x(2,3)=0$ $x(2,4)=0$
Nodo 3: $x(3,1)=0$ $x(3,5)=0$
Nodo 4: $x(4,3)=0$ $x(4,5)=0$

Iterazione: 1

Vertice 1 con eccesso 3

Il vertice 1 diventa permanente e viene espanso:
Aggiornamento etichetta vertice 2: label=1, pred=1
Aggiornamento etichetta vertice 4: label=4, pred=1

Il vertice 2 diventa permanente e viene espanso:
Aggiornamento etichetta vertice 3: label=5, pred=2
Vertice 4 non viene aggiornato

Il vertice 4 diventa permanente e viene espanso:
Vertice 3 non viene aggiornato
Aggiornamento etichetta vertice 5: label=7, pred=4

Il vertice 3 diventa permanente e viene espanso:
Vertice 1 non viene aggiornato
Vertice 5 non viene aggiornato

Il vertice 5 diventa permanente e viene espanso:

Riepilogo Cammini Minimi:

Cammino da 1 a 1 (costo=0): 1
Cammino da 1 a 2 (costo=1): 1 2
Cammino da 1 a 3 (costo=5): 1 2 3
Cammino da 1 a 4 (costo=4): 1 4
Cammino da 1 a 5 (costo=7): 1 4 5

Potenziali:

pi(1) = 0
pi(2) = -1
pi(3) = -5
pi(4) = -4
pi(5) = -7

Aumenta il flusso di delta=3

Iterazione: 2

Vertice 2 con eccesso 3

Il vertice 2 diventa permanente e viene espanso:
Aggiornamento etichetta vertice 3: label=0, pred=2
Aggiornamento etichetta vertice 4: label=2, pred=2

Il vertice 3 diventa permanente e viene espanso:
Aggiornamento etichetta vertice 1: label=7, pred=3
Aggiornamento etichetta vertice 5: label=4, pred=3

Il vertice 4 diventa permanente e viene espanso:
Aggiornamento etichetta vertice 1: label=2, pred=4
Vertice 3 non viene aggiornato
Aggiornamento etichetta vertice 5: label=2, pred=4

Il vertice 1 diventa permanente e viene espanso:
Vertice 2 non viene aggiornato
Vertice 4 non viene aggiornato

Il vertice 5 diventa permanente e viene espanso:
Vertice 4 non viene aggiornato

Riepilogo Cammini Minimi:

Cammino da 2 a 1 (costo=2): 2 4 1
Cammino da 2 a 2 (costo=0): 2
Cammino da 2 a 3 (costo=0): 2 3
Cammino da 2 a 4 (costo=2): 2 4
Cammino da 2 a 5 (costo=2): 2 4 5

Potenziali:

pi(1) = -2
pi(2) = -1
pi(3) = -5
pi(4) = -6
pi(5) = -9

Aumenta il flusso di delta=2

Iterazione: 3

Vertice 2 con eccesso 1

Il vertice 2 diventa permanente e viene espanso:
Aggiornamento etichetta vertice 3: label=0, pred=2
Aggiornamento etichetta vertice 4: label=0, pred=2

Il vertice 3 diventa permanente e viene espanso:
Aggiornamento etichetta vertice 1: label=5, pred=3
Aggiornamento etichetta vertice 5: label=2, pred=3

Il vertice 4 diventa permanente e viene espanso:
Aggiornamento etichetta vertice 1: label=0, pred=4
Vertice 2 non viene aggiornato
Vertice 3 non viene aggiornato
Vertice 5 non viene aggiornato

Il vertice 1 diventa permanente e viene espanso:
Vertice 2 non viene aggiornato
Vertice 4 non viene aggiornato

Il vertice 5 diventa permanente e viene espanso:
Vertice 4 non viene aggiornato

Riepilogo Cammini Minimi:
Cammino da 2 a 1 (costo=0): 2 4 1
Cammino da 2 a 2 (costo=0): 2
Cammino da 2 a 3 (costo=0): 2 3
Cammino da 2 a 4 (costo=0): 2 4
Cammino da 2 a 5 (costo=2): 2 3 5

Potenziali:

pi(1) = -2
pi(2) = -1
pi(3) = -5
pi(4) = -6
pi(5) = -11

Aumenta il flusso di delta=1

Numero Iterazioni: 3

Risultato Flusso di Costo Minimo
Nodo 1: x(1,2)=0 x(1,4)=3
Nodo 2: x(2,3)=1 x(2,4)=2
Nodo 3: x(3,1)=0 x(3,5)=1
Nodo 4: x(4,3)=0 x(4,5)=5

Costo Soluzione = 47

- e) Determinare i cammini di costo minimo dal vertice 1 a tutti gli altri vertici utilizzando l'algoritmo di Dijkstra e i costi c_{ij} .

Algoritmo Cammini di Costo Minimo: Dijkstra

Il vertice 1 diventa permanente e viene espanso:
Aggiornamento etichetta vertice 2: label=1, pred=1
Aggiornamento etichetta vertice 4: label=4, pred=1

Il vertice 2 diventa permanente e viene espanso:
Aggiornamento etichetta vertice 3: label=5, pred=2
Vertice 4 non viene aggiornato

Il vertice 4 diventa permanente e viene espanso:
Vertice 3 non viene aggiornato
Aggiornamento etichetta vertice 5: label=7, pred=4

Il vertice 3 diventa permanente e viene espanso:
Vertice 1 non viene aggiornato
Vertice 5 non viene aggiornato

Il vertice 5 diventa permanente e viene espanso:
Non ha successori

Riepilogo Cammini Minimi:
Cammino da 1 a 2 (costo=1): 1 2
Cammino da 1 a 3 (costo=5): 1 2 3
Cammino da 1 a 4 (costo=4): 1 4
Cammino da 1 a 5 (costo=7): 1 4 5

- f) Determinare i cammini di costo minimo dal vertice 1 a tutti gli altri vertici utilizzando l'algoritmo di Bellman-Ford e i costi c_{ij} .

Algoritmo Cammini di Costo Minimo: Bellman-Ford

Iterazione 1:

```
L'arco (1,2) viola la condizione: Label[2]=1000 > Label[1]=0 + Costo(1,2)=1
==> Label[2]=1 e pred[2]=0
L'arco (2,3) viola la condizione: Label[3]=1000 > Label[2]=1 + Costo(2,3)=4
==> Label[3]=5 e pred[3]=1
L'arco (1,4) viola la condizione: Label[4]=1000 > Label[1]=0 + Costo(1,4)=4
==> Label[4]=4 e pred[4]=0
L'arco (3,5) viola la condizione: Label[5]=1000 > Label[3]=5 + Costo(3,5)=6
==> Label[5]=11 e pred[5]=2
L'arco (4,5) viola la condizione: Label[5]=11 > Label[4]=4 + Costo(4,5)=3
==> Label[5]=7 e pred[5]=3
```

Iterazione 2:

Non ci sono violazioni

Riepilogo Cammini Minimi:

Cammino da 1 a 2 (costo=1):	1	2	
Cammino da 1 a 3 (costo=5):	1	2	3
Cammino da 1 a 4 (costo=4):	1	4	
Cammino da 1 a 5 (costo=7):	1	4	5

- g) Determinare i cammini di costo minimo per tutte le coppie di vertici utilizzando i costi c_{ij} .

[Nota: in questo caso all'esame potremo chiedere svolgere solo la prima iterazione oppure le prime due.]

Algoritmo Cammini di Costo Minimo: Floyd-Warshal

Iterazione 0:

Matrice u_ij:

0	1	1000	4	1000
1000	0	4	5	1000
2	3	0	6	6
1000	1000	3	0	3
1000	1000	1000	1000	0

Matrice Pred_ij:

1	1	1	1	1
2	2	2	2	2
3	1	3	1	3
4	4	4	4	4
5	5	5	5	5

Iterazione 1:

Matrice u_ij:

0	1	5	4	1000
1000	0	4	5	1000
2	3	0	6	6
1000	1000	3	0	3
1000	1000	1000	1000	0

Matrice Pred_ij:

1	1	2	1	1
2	2	2	2	2
3	1	3	1	3
4	4	4	4	4
5	5	5	5	5

Iterazione 2:

Matrice u_ij:

0	1	5	4	11
6	0	4	5	10
2	3	0	6	6
5	6	3	0	3
1000	1000	1000	1000	0

Matrice Pred_ij:

1	1	2	1	3
3	2	2	2	3
3	1	3	1	3
3	1	4	4	4
5	5	5	5	5

Iterazione 3:

Matrice u_ij:

0	1	5	4	7
6	0	4	5	8
2	3	0	6	6
5	6	3	0	3
1000	1000	1000	1000	0

Matrice Pred_ij:

1	1	2	1	4
3	2	2	2	4
3	1	3	1	3
3	1	4	4	4
5	5	5	5	5

Iterazione 4:

Matrice u_ij:

0	1	5	4	7
6	0	4	5	8
2	3	0	6	6
5	6	3	0	3
1000	1000	1000	1000	0

Matrice Pred_ij:

1	1	2	1	4
3	2	2	2	4
3	1	3	1	3
3	1	4	4	4
5	5	5	5	5

- h) Determinare l'albero di copertura di costo minimo sostituendo gli archi (i, j) con i lati $\{i, j\}$ e utilizzando i costi c_{ij} .

Alberi di Copertura di Costo Minimo: Algoritmo Kruskal

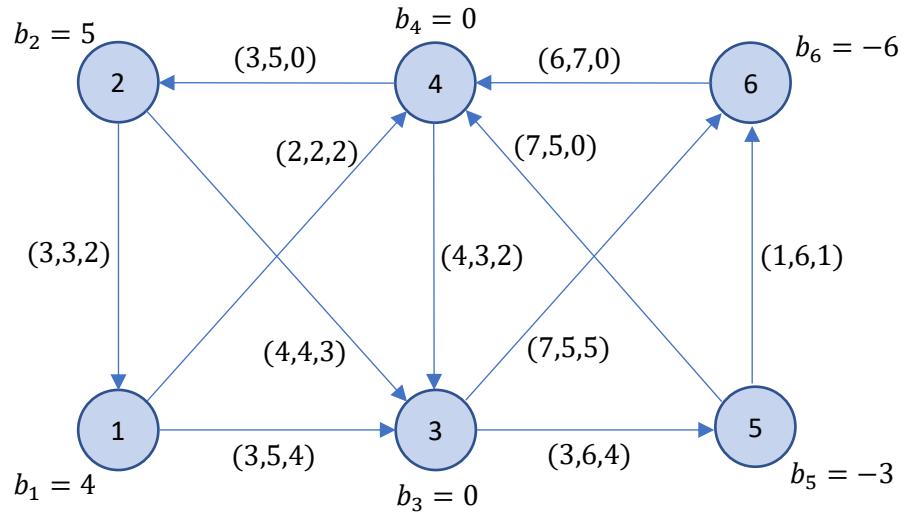
Selezione dei lati:

- 1) Inserimento del lato {1,2}
 - 2) Inserimento del lato {3,1}
 - 3) Inserimento del lato {4,3}
 - 4) Inserimento del lato {4,5}
- Costo = 9

Alberi di Copertura di Costo Minimo: Algoritmo Prim

- 1) Inserimento del lato {1, 2}
 - 2) Inserimento del lato {1, 3}
 - 3) Inserimento del lato {3, 4}
 - 4) Inserimento del lato {4, 5}
- Costo = 9

2) Si consideri il seguente grafo G:



Su ogni arco (i,j) è riportata la tripletta (c_{ij}, u_{ij}, x_{ij}) , dove c_{ij} è il costo per trasportare una unità di flusso, u_{ij} è la capacità e x_{ij} è il flusso corrente.

a) Determinare il flusso massimo partendo dal flusso nullo (i.e., $x_{ij} = 0$, per ogni arco (i,j)).

Algoritmo Flusso Massimo

Grafo Iniziale Flusso Massimo

Nodo 1: $x(1,3)=0 \quad x(1,4)=0$
 Nodo 2: $x(2,1)=0 \quad x(2,3)=0$
 Nodo 3: $x(3,5)=0 \quad x(3,6)=0$
 Nodo 4: $x(4,2)=0 \quad x(4,3)=0$
 Nodo 5: $x(5,4)=0 \quad x(5,6)=0$

Etichetta nodo 1: [1,1000]
 Etichetta nodo 3: [1,5]
 Etichetta nodo 4: [1,2]
 Etichetta nodo 5: [3,5]
 Etichetta nodo 6: [3,5]
 Etichetta nodo 2: [4,2]
 Aumenta il flusso di 5 nel cammino: (1,3) (3,6)
 Etichetta nodo 1: [1,1000]
 Etichetta nodo 4: [1,2]
 Etichetta nodo 2: [4,2]
 Etichetta nodo 3: [4,2]
 Etichetta nodo 5: [3,2]
 Etichetta nodo 6: [5,2]
 Aumenta il flusso di 2 nel cammino: (1,4) (4,3) (3,5) (5,6)
 Etichetta nodo 1: [1,1000]

Flusso Massimo Trovato!

Numero Iterazioni: 3

Risultato Flusso Massimo
 Nodo 1: $x(1,3)=5 \quad x(1,4)=2$
 Nodo 2: $x(2,1)=0 \quad x(2,3)=0$
 Nodo 3: $x(3,5)=2 \quad x(3,6)=5$
 Nodo 4: $x(4,2)=0 \quad x(4,3)=2$
 Nodo 5: $x(5,4)=0 \quad x(5,6)=2$

Flusso Massimo = 7

b) Determinare il flusso di costo minimo con l'algoritmo Cycle Cancelling generando una soluzione ammissibile (e non utilizzando i flussi x_{ij} forniti nel testo). [Nota: in questo caso

all'esame potremo chiedere di generare solo la soluzione ammissibile, oppure svolgere solo la prima iterazione oppure le prime due

Algoritmo Flusso di Costo Minimo: senza soluzione ammissibile

Cerca Flusso Ammissibile
Etichetta nodo 7: [7,1000]
Etichetta nodo 1: [7,4]
Etichetta nodo 2: [7,5]
Etichetta nodo 3: [1,4]
Etichetta nodo 4: [1,2]
Etichetta nodo 5: [3,4]
Etichetta nodo 6: [3,4]
Etichetta nodo 8: [5,3]
Aumenta il flusso di 3 nel cammino: (7,1) (1,3) (3,5) (5,8)
Etichetta nodo 7: [7,1000]
Etichetta nodo 1: [7,1]
Etichetta nodo 2: [7,5]
Etichetta nodo 3: [1,1]
Etichetta nodo 4: [1,1]
Etichetta nodo 5: [3,1]
Etichetta nodo 6: [3,1]
Etichetta nodo 8: [6,1]
Aumenta il flusso di 1 nel cammino: (7,1) (1,3) (3,6) (6,8)
Etichetta nodo 7: [7,1000]
Etichetta nodo 2: [7,5]
Etichetta nodo 1: [2,3]
Etichetta nodo 3: [2,4]
Etichetta nodo 4: [1,2]
Etichetta nodo 5: [3,3]
Etichetta nodo 6: [3,4]
Etichetta nodo 8: [6,4]
Aumenta il flusso di 4 nel cammino: (7,2) (2,3) (3,6) (6,8)
Etichetta nodo 7: [7,1000]
Etichetta nodo 2: [7,1]
Etichetta nodo 1: [2,1]
Etichetta nodo 3: [1,1]
Etichetta nodo 4: [1,1]
Etichetta nodo 5: [3,1]
Etichetta nodo 6: [5,1]
Etichetta nodo 8: [6,1]
Aumenta il flusso di 1 nel cammino: (7,2) (2,1) (1,3) (3,5) (5,6) (6,8)
Etichetta nodo 7: [7,1000]

Flusso Massimo Trovato!

Numero Iterazioni: 5

Soluzione Ammissibile Iniziale
Nodo 1: $x(1,3)=5$ $x(1,4)=0$
Nodo 2: $x(2,1)=1$ $x(2,3)=4$
Nodo 3: $x(3,5)=4$ $x(3,6)=5$
Nodo 4: $x(4,2)=0$ $x(4,3)=0$
Nodo 5: $x(5,4)=0$ $x(5,6)=1$

Cerca Cicli di Costo Negativo

Iterazione: 1
Ciclo: $C(3,5)=3$ $C(5,6)=1$ $C(6,3)=-7$
Aumento flusso di: 2

Iterazione: 2
Ciclo non trovato: Flusso Costo Minimo Trovato

Numero Iterazioni: 2

Risultato Flusso di Costo Minimo
Nodo 1: $x(1,3)=5$ $x(1,4)=0$
Nodo 2: $x(2,1)=1$ $x(2,3)=4$
Nodo 3: $x(3,5)=6$ $x(3,6)=3$
Nodo 4: $x(4,2)=0$ $x(4,3)=0$
Nodo 5: $x(5,4)=0$ $x(5,6)=3$

Costo Soluzione = 76

- c) Determinare il flusso di costo minimo con l'algoritmo Cycle Cancelling partendo dai flussi x_{ij} forniti nel testo. [Nota: in questo caso all'esame potremo chiedere svolgere solo la prima iterazione oppure le prime due]

Algoritmo Flusso di Costo Minimo: con soluzione ammissibile

Soluzione Ammissibile Iniziale
Nodo 1: $x(1,3)=4$ $x(1,4)=2$
Nodo 2: $x(2,1)=2$ $x(2,3)=3$
Nodo 3: $x(3,5)=4$ $x(3,6)=5$
Nodo 4: $x(4,2)=0$ $x(4,3)=2$
Nodo 5: $x(5,4)=0$ $x(5,6)=1$

Cerca Cicli di Costo Negativo

Iterazione: 1
Ciclo: $C(1,2)=-3$ $C(2,3)=4$ $C(3,1)=-3$
Aumento flusso di: 1

Iterazione: 2
Ciclo: $C(1,3)=3$ $C(3,4)=-4$ $C(4,1)=-2$
Aumento flusso di: 2

Iterazione: 3
Ciclo: $C(3,5)=3$ $C(5,6)=1$ $C(6,3)=-7$
Aumento flusso di: 2

Iterazione: 4
Ciclo non trovato: Flusso Costo Minimo Trovato

Numero Iterazioni: 4

Risultato Flusso di Costo Minimo
Nodo 1: $x(1,3)=5$ $x(1,4)=0$
Nodo 2: $x(2,1)=1$ $x(2,3)=4$
Nodo 3: $x(3,5)=6$ $x(3,6)=3$
Nodo 4: $x(4,2)=0$ $x(4,3)=0$
Nodo 5: $x(5,4)=0$ $x(5,6)=3$

Costo Soluzione = 76

- d) Determinare il flusso di costo minimo con l'algoritmo Successive Shortest Path partendo dai flussi x_{ij} nulli e potenziali π_i nulli. [Nota: in questo caso all'esame potremo chiedere svolgere solo la prima iterazione oppure le prime due. In alternativa potremo fornire il flusso e i potenziali e chiedere di svolgere una o due iterazioni]

Algoritmo Flusso di Costo Minimo: Successive Shortest Path

Soluzione Iniziale
Nodo 1: $x(1,3)=0$ $x(1,4)=0$
Nodo 2: $x(2,1)=0$ $x(2,3)=0$
Nodo 3: $x(3,5)=0$ $x(3,6)=0$
Nodo 4: $x(4,2)=0$ $x(4,3)=0$
Nodo 5: $x(5,4)=0$ $x(5,6)=0$

Iterazione: 1

Vertice 1 con eccesso 4

Il vertice 1 diventa permanente e viene espanso:
Aggiornamento etichetta vertice 3: label=3, pred=1
Aggiornamento etichetta vertice 4: label=2, pred=1

Il vertice 4 diventa permanente e viene espanso:
Aggiornamento etichetta vertice 2: label=5, pred=4
Vertice 3 non viene aggiornato

```
Il vertice 3 diventa permanente e viene espanso:  
Aggiornamento etichetta vertice 5: label=6, pred=3  
Aggiornamento etichetta vertice 6: label=10, pred=3
```

```
Il vertice 2 diventa permanente e viene espanso:  
Vertice 1 non viene aggiornato  
Vertice 3 non viene aggiornato
```

```
Il vertice 5 diventa permanente e viene espanso:  
Vertice 4 non viene aggiornato  
Aggiornamento etichetta vertice 6: label=7, pred=5
```

```
Il vertice 6 diventa permanente e viene espanso:  
Vertice 4 non viene aggiornato
```

Riepilogo Cammini Minimi:

```
Cammino da 1 a 1 (costo=0): 1  
Cammino da 1 a 2 (costo=5): 1 4 2  
Cammino da 1 a 3 (costo=3): 1 3  
Cammino da 1 a 4 (costo=2): 1 4  
Cammino da 1 a 5 (costo=6): 1 3 5  
Cammino da 1 a 6 (costo=7): 1 3 5 6
```

Potenziali:

```
pi(1) = 0  
pi(2) = -5  
pi(3) = -3  
pi(4) = -2  
pi(5) = -6  
pi(6) = -7
```

```
Aumenta il flusso di delta=3
```

```
Iterazione: 2
```

```
Vertice 1 con eccesso 1
```

```
Il vertice 1 diventa permanente e viene espanso:  
Aggiornamento etichetta vertice 3: label=0, pred=1  
Aggiornamento etichetta vertice 4: label=0, pred=1
```

```
Il vertice 3 diventa permanente e viene espanso:  
Vertice 1 non viene aggiornato  
Aggiornamento etichetta vertice 5: label=0, pred=3  
Aggiornamento etichetta vertice 6: label=3, pred=3
```

```
Il vertice 4 diventa permanente e viene espanso:  
Aggiornamento etichetta vertice 2: label=0, pred=4  
Vertice 3 non viene aggiornato
```

```
Il vertice 2 diventa permanente e viene espanso:  
Vertice 1 non viene aggiornato  
Vertice 3 non viene aggiornato
```

```
Il vertice 5 diventa permanente e viene espanso:  
Vertice 3 non viene aggiornato  
Vertice 4 non viene aggiornato  
Aggiornamento etichetta vertice 6: label=0, pred=5
```

```
Il vertice 6 diventa permanente e viene espanso:  
Vertice 4 non viene aggiornato
```

Riepilogo Cammini Minimi:

```
Cammino da 1 a 1 (costo=0): 1  
Cammino da 1 a 2 (costo=0): 1 4 2  
Cammino da 1 a 3 (costo=0): 1 3  
Cammino da 1 a 4 (costo=0): 1 4  
Cammino da 1 a 5 (costo=0): 1 3 5  
Cammino da 1 a 6 (costo=0): 1 3 5 6
```

Potenziali:

```
pi(1) = 0  
pi(2) = -5  
pi(3) = -3
```

```

pi(4) = -2
pi(5) = -6
pi(6) = -7

Aumenta il flusso di delta=1

Iterazione: 3

Vertice 2 con eccesso 5

Il vertice 2 diventa permanente e viene espanso:
    Aggiornamento etichetta vertice 1: label=8, pred=2
    Aggiornamento etichetta vertice 3: label=6, pred=2

Il vertice 3 diventa permanente e viene espanso:
    Aggiornamento etichetta vertice 1: label=6, pred=3
    Aggiornamento etichetta vertice 5: label=6, pred=3
    Aggiornamento etichetta vertice 6: label=9, pred=3

Il vertice 1 diventa permanente e viene espanso:
    Vertice 3 non viene aggiornato
    Aggiornamento etichetta vertice 4: label=6, pred=1

Il vertice 4 diventa permanente e viene espanso:
    Vertice 2 non viene aggiornato
    Vertice 3 non viene aggiornato

Il vertice 5 diventa permanente e viene espanso:
    Vertice 3 non viene aggiornato
    Vertice 4 non viene aggiornato
    Aggiornamento etichetta vertice 6: label=6, pred=5

Il vertice 6 diventa permanente e viene espanso:
    Vertice 4 non viene aggiornato
    Vertice 5 non viene aggiornato

Riepilogo Cammini Minimi:
    Cammino da 2 a 1 (costo=6): 2 3 1
    Cammino da 2 a 2 (costo=0): 2
    Cammino da 2 a 3 (costo=6): 2 3
    Cammino da 2 a 4 (costo=6): 2 3 1 4
    Cammino da 2 a 5 (costo=6): 2 3 5
    Cammino da 2 a 6 (costo=6): 2 3 5 6

Potenziali:
pi(1) = -6
pi(2) = -5
pi(3) = -9
pi(4) = -8
pi(5) = -12
pi(6) = -13

Aumenta il flusso di delta=2

Iterazione: 4

Vertice 2 con eccesso 3

Il vertice 2 diventa permanente e viene espanso:
    Aggiornamento etichetta vertice 1: label=2, pred=2
    Aggiornamento etichetta vertice 3: label=0, pred=2

Il vertice 3 diventa permanente e viene espanso:
    Aggiornamento etichetta vertice 1: label=0, pred=3
    Vertice 2 non viene aggiornato
    Aggiornamento etichetta vertice 5: label=1000, pred=3
    Aggiornamento etichetta vertice 6: label=3, pred=3

Il vertice 1 diventa permanente e viene espanso:
    Vertice 3 non viene aggiornato
    Aggiornamento etichetta vertice 4: label=0, pred=1

Il vertice 4 diventa permanente e viene espanso:
    Vertice 2 non viene aggiornato

```

Vertice 3 non viene aggiornato

Il vertice 6 diventa permanente e viene espanso:

Vertice 4 non viene aggiornato

Aggiornamento etichetta vertice 5: label=3, pred=6

Il vertice 5 diventa permanente e viene espanso:

Vertice 3 non viene aggiornato

Vertice 4 non viene aggiornato

Vertice 6 non viene aggiornato

Riepilogo Cammini Minimi:

Cammino da 2 a 1 (costo=0): 2 3 1

Cammino da 2 a 2 (costo=0): 2

Cammino da 2 a 3 (costo=0): 2 3

Cammino da 2 a 4 (costo=0): 2 3 1 4

Cammino da 2 a 5 (costo=3): 2 3 6 5

Cammino da 2 a 6 (costo=3): 2 3 6

Potenziali:

pi(1) = -6

pi(2) = -5

pi(3) = -9

pi(4) = -8

pi(5) = -15

pi(6) = -16

Aumenta il flusso di delta=2

Iterazione: 5

Vertice 2 con eccesso 1

Il vertice 2 diventa permanente e viene espanso:

Aggiornamento etichetta vertice 1: label=2, pred=2

Aggiornamento etichetta vertice 3: label=1000, pred=2

Il vertice 1 diventa permanente e viene espanso:

Aggiornamento etichetta vertice 3: label=2, pred=1

Aggiornamento etichetta vertice 4: label=2, pred=1

Il vertice 3 diventa permanente e viene espanso:

Vertice 1 non viene aggiornato

Vertice 2 non viene aggiornato

Aggiornamento etichetta vertice 5: label=1002, pred=3

Aggiornamento etichetta vertice 6: label=2, pred=3

Il vertice 4 diventa permanente e viene espanso:

Vertice 2 non viene aggiornato

Vertice 3 non viene aggiornato

Il vertice 6 diventa permanente e viene espanso:

Vertice 3 non viene aggiornato

Vertice 4 non viene aggiornato

Aggiornamento etichetta vertice 5: label=2, pred=6

Il vertice 5 diventa permanente e viene espanso:

Vertice 3 non viene aggiornato

Vertice 4 non viene aggiornato

Vertice 6 non viene aggiornato

Riepilogo Cammini Minimi:

Cammino da 2 a 1 (costo=2): 2 1

Cammino da 2 a 2 (costo=0): 2

Cammino da 2 a 3 (costo=2): 2 1 3

Cammino da 2 a 4 (costo=2): 2 1 4

Cammino da 2 a 5 (costo=2): 2 1 3 6 5

Cammino da 2 a 6 (costo=2): 2 1 3 6

Potenziali:

pi(1) = -8

pi(2) = -5

pi(3) = -11

pi(4) = -10

```

pi(5) = -17
pi(6) = -18

Aumenta il flusso di delta=1

Numero Iterazioni: 5

Risultato Flusso di Costo Minimo
Nodo 1: x(1,3)=5 x(1,4)=0
Nodo 2: x(2,1)=1 x(2,3)=4
Nodo 3: x(3,5)=6 x(3,6)=3
Nodo 4: x(4,2)=0 x(4,3)=0
Nodo 5: x(5,4)=0 x(5,6)=3

Costo Soluzione = 76

```

- e) Determinare i cammini di costo minimo dal vertice 1 a tutti gli altri vertici utilizzando l'algoritmo di Dijkstra e i costi c_{ij} .

Algoritmo Cammini di Costo Minimo: Dijkstra

Il vertice 1 diventa permanente e viene espanso:
 Aggiornamento etichetta vertice 3: label=3, pred=1
 Aggiornamento etichetta vertice 4: label=2, pred=1

Il vertice 4 diventa permanente e viene espanso:
 Aggiornamento etichetta vertice 2: label=5, pred=4
 Vertice 3 non viene aggiornato

Il vertice 3 diventa permanente e viene espanso:
 Aggiornamento etichetta vertice 5: label=6, pred=3
 Aggiornamento etichetta vertice 6: label=10, pred=3

Il vertice 2 diventa permanente e viene espanso:
 Vertice 1 non viene aggiornato
 Vertice 3 non viene aggiornato

Il vertice 5 diventa permanente e viene espanso:
 Vertice 4 non viene aggiornato
 Aggiornamento etichetta vertice 6: label=7, pred=5

Il vertice 6 diventa permanente e viene espanso:
 Vertice 4 non viene aggiornato

Riepilogo Cammini Minimi:

Cammino da 1 a 2 (costo=5):	1	4	2	
Cammino da 1 a 3 (costo=3):	1	3		
Cammino da 1 a 4 (costo=2):	1	4		
Cammino da 1 a 5 (costo=6):	1	3	5	
Cammino da 1 a 6 (costo=7):	1	3	5	6

- f) Determinare i cammini di costo minimo dal vertice 1 a tutti gli altri vertici utilizzando l'algoritmo di Bellman-Ford e i costi c_{ij} .

Algoritmo Cammini di Costo Minimo: Bellman-Ford

Iterazione 1:
 L'arco (1,3) viola la condizione: Label[3]=1000 > Label[1]=0 + Costo(1,3)=3
 \Rightarrow Label[3]=3 e pred[3]=0
 L'arco (1,4) viola la condizione: Label[4]=1000 > Label[1]=0 + Costo(1,4)=2
 \Rightarrow Label[4]=2 e pred[4]=0
 L'arco (3,5) viola la condizione: Label[5]=1000 > Label[3]=3 + Costo(3,5)=3
 \Rightarrow Label[5]=6 e pred[5]=2
 L'arco (3,6) viola la condizione: Label[6]=1000 > Label[3]=3 + Costo(3,6)=7
 \Rightarrow Label[6]=10 e pred[6]=2
 L'arco (5,6) viola la condizione: Label[6]=10 > Label[5]=6 + Costo(5,6)=1
 \Rightarrow Label[6]=7 e pred[6]=4

Iterazione 2:
 L'arco (4,2) viola la condizione: Label[2]=1000 > Label[4]=2 + Costo(4,2)=3
 \Rightarrow Label[2]=5 e pred[2]=3

```

Iterazione 3:
.. Non ci sono violazioni
Riepilogo Cammini Minimi:
Cammino da 1 a 2 (costo=5): 1 4 2
Cammino da 1 a 3 (costo=3): 1 3
Cammino da 1 a 4 (costo=2): 1 4
Cammino da 1 a 5 (costo=6): 1 3 5
Cammino da 1 a 6 (costo=7): 1 3 5 6

```

- g) Determinare i cammini di costo minimo per tutte le coppie di vertici utilizzando i costi c_{ij} .
[Nota: in questo caso all'esame potremo chiedere svolgere solo la prima iterazione oppure le prime due.]

Algoritmo Cammini di Costo Minimo: Floyd-Warshal

Iterazione 0:

```

Matrice u_ij:
 0 1000   3   2 1000 1000
 3   0   4   5 1000 1000
1000 1000   0 1000   3   7
1000   3   4   0 1000 1000
1000 1000 1000   7   0   1
1000 1000 1000   6 1000   0

```

```

Matrice Pred_ij:
 1   1   1   1   1   1
 2   2   2   1   2   2
 3   3   3   3   3   3
 4   4   4   4   4   4
 5   5   5   5   5   5
 6   6   6   6   6   6

```

Iterazione 1:

```

Matrice u_ij:
 0 1000   3   2 1000 1000
 3   0   4   5 1000 1000
1000 1000   0 1000   3   7
 6   3   4   0 1000 1000
1000 1000 1000   7   0   1
1000 1000 1000   6 1000   0

```

```

Matrice Pred_ij:
 1   1   1   1   1   1
 2   2   2   1   2   2
 3   3   3   3   3   3
 2   4   4   4   4   4
 5   5   5   5   5   5
 6   6   6   6   6   6

```

Iterazione 2:

```

Matrice u_ij:
 0 1000   3   2   6   10
 3   0   4   5   7   11
1000 1000   0 1000   3   7
 6   3   4   0   7   11
1000 1000 1000   7   0   1
1000 1000 1000   6 1000   0

```

```

Matrice Pred_ij:
 1   1   1   1   3   3
 2   2   2   1   3   3
 3   3   3   3   3   3
 2   4   4   4   3   3
 5   5   5   5   5   5
 6   6   6   6   6   6

```

Iterazione 3:

Matrice u_ij:

0	5	3	2	6	10
3	0	4	5	7	11
1000	1000	0	1000	3	7
6	3	4	0	7	11
13	10	11	7	0	1
12	9	10	6	13	0

Matrice Pred_ij:

1	4	1	1	3	3
2	2	2	1	3	3
3	3	3	3	3	3
2	4	4	4	3	3
2	4	4	5	5	5
2	4	4	6	3	6

Iterazione 4:

Matrice u_ij:

0	5	3	2	6	7
3	0	4	5	7	8
16	13	0	10	3	4
6	3	4	0	7	8
13	10	11	7	0	1
12	9	10	6	13	0

Matrice Pred_ij:

1	4	1	1	3	5
2	2	2	1	3	5
2	4	3	5	3	5
2	4	4	4	3	5
2	4	4	5	5	5
2	4	4	6	3	6

Iterazione 5:

Matrice u_ij:

0	5	3	2	6	7
3	0	4	5	7	8
16	13	0	10	3	4
6	3	4	0	7	8
13	10	11	7	0	1
12	9	10	6	13	0

Matrice Pred_ij:

1	4	1	1	3	5
2	2	2	1	3	5
2	4	3	5	3	5
2	4	4	4	3	5
2	4	4	5	5	5
2	4	4	6	3	6

- h) Determinare l'albero di copertura di costo minimo sostituendo gli archi (i,j) con i lati $\{i,j\}$ e utilizzando i costi c_{ij} .

Alberi di Copertura di Costo Minimo: Algoritmo Kruskal

Selezione dei lati:

- 1) Inserimento del lato {5,6}
 - 2) Inserimento del lato {1,4}
 - 3) Inserimento del lato {1,3}
 - 4) Inserimento del lato {2,1}
 - 5) Inserimento del lato {3,5}
- Costo = 12

Alberi di Copertura di Costo Minimo: Algoritmo Prim

- 1) Inserimento del lato {1, 4}
 - 2) Inserimento del lato {1, 2}
 - 3) Inserimento del lato {1, 3}
 - 4) Inserimento del lato {3, 5}
 - 5) Inserimento del lato {5, 6}
- Costo = 12