Università degli Studi di Padova

Algoritmi Avanzati

A.A. 2019/2020

29 giugno 2020, 10:00–12:00

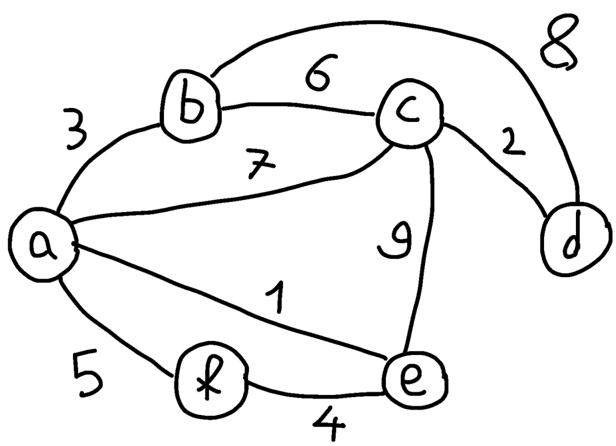
Prima Parte: domande di teoria

Domanda 1 (6 punti) Si consideri il seguente grafo pesato, rappresentato tramite una matrice di adiacenza dove ogni valore numerico rappresenta il peso del lato corrispondente, e dove il simbolo '$-$' indica l'assenza del lato tra i nodi corrispondenti.

	a	b	c	d	e	f
a	-	3	7	-	1	5
b		-	6	8	-	-
c			-	2	9	-
d				-	-	-
e					-	4
f						-

1. Disegnare il grafo.
2. Elencare i lati del minimum spanning tree nell'ordine in cui sono selezionati dall'algoritmo di Kruskal.
3. Elencare i lati del minimum spanning tree nell'ordine in cui sono selezionati dall'algoritmo di Prim a partire dal nodo a.

Soluzione:

1. 

2. $(a, e), (c, d), (a, b), (e, f), (b, c)$.
3. $(a, e), (a, b), (e, f), (b, c), (c, d)$.

Domanda 2 (7 punti) Con riferimento al problema del minimum vertex cover, che durante il corso abbiamo visto come 2-approssimare attraverso il calcolo di un matching massimale:

1. Dare la definizione di vertex cover in un grafo.
2. Dare la definizione di matching in un grafo.
3. Determinare un matching massimale nel grafo della Domanda 1.

Soluzione:

1. Un vertex cover è un sottoinsieme C dei vertici tali che ogni lato ha almeno un'estremità in C.
2. Un matching è un insieme di lati disgiunti, cioè che non condividono estremi.
3. Per esempio $\{(a,b),(c,d),(e,f)\}$, oppure $\{(a,c),(b,d),(e,f)\}$.

Seconda Parte: risoluzione di problemi

Esercizio 1 (11 punti) Sia dato un insieme S di n numeri interi e un ulteriore intero t, e si assuma che $\forall s \in S, 0 \leq s \leq t$. Si consideri il problema di ottimizzazione dove l'insieme delle soluzioni ammissibili è

$$\{S' \subseteq S \,\text{tale che} \sum_{s \in S'} s \leq t\},$$

il costo di una soluzione ammissibile S' è $c(S') = \sum_{s \in S'} s$, e si richiede di calcolare il costo massimo tra tutti i costi delle soluzioni ammissibili.

1. Progettare un semplice algoritmo polinomiale di 2-approssimazione per il problema. (Suggerimento: si consideri un ordinamento *decrescente* dei valori di S, e poi si faccia una singola passata su tali valori.)
2. Dimostrare che l'algoritmo trovato è un algoritmo di 2-approssimazione.

Soluzione:

1.
```
APPROX_SS(S, t)
{s_1, s_2,..., s_n} <- SORT-DECREASING(S)
sum = s_1
for i = 2 to n do
   if sum + s_i <= t then
      sum = sum + s_i
   else
      return sum
return sum
```

2. Prima di tutto osserviamo che, siccome sum è inizializzato a $s_1 \leq t$ e un valore s_i è aggiunto a *sum* solo se $sum + s_i \leq t$, il valore ritornato è sempre il costo di una soluzione ammissibile. Se $s^\star$ denota il costo massimo, ora dobbiamo dimostrare che $s^\star / sum \leq 2$.

   caso 1 L'algoritmo ritorna fuori dal ciclo for: quindi $sum = \sum_{s \in S} s \leq t$, cioè $sum = s^\star$, e quindi $s^\star / sum = 1 \leq 2$.

caso 2 L'algoritmo ritorna da dentro al ciclo for: quindi esiste un indice i' tale che $sum + s_{i'} > t$. Si osservi che

$$s_{i'} < s_1 \leq sum,$$

e quindi

$$2 \cdot sum > sum + s_{i'} > t,$$

ovvero

$$sum > \frac{t}{2} \geq \frac{s^\star}{2}.$$

Esercizio 2 (9 punti) Siano $X_1, X_2, \ldots, X_n$ variabili indicatore indipendenti con $Pr(X_i = 1) = 1/(4e)$. Sia $X = \sum_{i=1}^n X_i$ e $\mu = E[X]$. Applicando il seguente bound di Chernoff, che vale per ogni $\delta > 0$,

$$Pr(X > (1+\delta)\mu) < \left(\frac{e^\delta}{(1+\delta)^{1+\delta}}\right)^\mu$$

si dimostri che

$$Pr(X > n/2) < \frac{1}{(\sqrt{2})^n}.$$

Soluzione: Per applicare il bound di Chernoff poniamo $n/2$ uguale a $(1+\delta)\mu$; siccome $\mu = E[X] = \sum_{i=1}^n E[X_i] = n/(4e)$, otteniamo $\delta = 2e - 1$. Quindi

$$\begin{aligned} Pr(X > n/2) &= Pr(X > (1 + 2e - 1)\mu) \\ &< \left(\frac{e^{2e-1}}{(2e)^{2e}}\right)^{n/(4e)} \\ &< 1/(2^{2e/4e})^n \\ &= \left(1/\sqrt{2}\right)^n. \end{aligned}$$