Università degli Studi di Padova

Algoritmi Avanzati

A.A. 2019/2020

29 giugno 2020, 10:00–12:00

First Part: Theory Questions

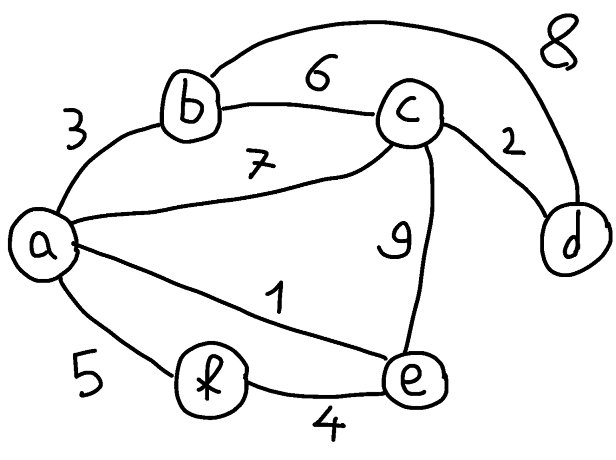
Question 1 (6 points) Consider the following weighted graph, represented by an adjacency matrix where each numerical value represents the weight of the corresponding edge, and where the symbol '$-$' indicates the absence of the edge between the corresponding nodes.

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

1. Draw the graph.
2. List the edges of the minimum spanning tree in the order they are selected by Kruskal's algorithm.
3. List the edges of the minimum spanning tree in the order they are selected by Prim's algorithm starting at node a.

Solution:

1.



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

Question 2 (7 points) With reference to the problem of the minimum vertex cover, which in class we have 2-approximated by computing a maximal matching:

1. Give the definition of vertex cover of a graph.
2. Give the definition of matching of a graph.
3. Find a maximal matching in the graph of Question 1.

Solution:

1. A vertex cover of a graph is a set of vertices that includes at least one endpoint of every edge of the graph.
2. A matching of a graph is a set of edges without common vertices.
3. For example $\{(a,b),(c,d),(e,f)\}$ or $\{(a,c),(b,d),(e,f)\}$.

Second Part: Problem Solving

Exercise 1 (11 points) Given a set S of n integers and an additional integer t, assume that $\forall s \in S, 0 \le s \le t$. Consider the optimization problem where the set of feasible solutions is

$$\{S' \subseteq S \,\text{such that} \sum_{s \in S'} s \le t\},$$

the cost of a feasible solution S' is $c(S') = \sum_{s \in S'} s$, and the goal is to compute the maximum cost among all the costs of the feasible solutions.

1. Design a simple 2-approximation polynomial-time algorithm for this problem. (Hint: consider a *descending* ordering of the values in S, and then do a single pass over such values.)
2. Prove that such algorithm is a 2-approximation algorithm.

Solution:

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. First of all we observe that, since *sum* is initialized to $s_1 \le t$ and a value s_i is added to *sum* only if $sum + s_i \le t$, the returned value is always the cost of a feasible solution. If $s^\star$ denotes the maximum cost, we now need to prove that $s^\star / sum \le 2$.

   case 1 The algorithm returns out of the for loop: hence $sum = \sum_{s \in S} s \le t$, that is $sum = s^\star$, and thus $s^\star / sum = 1 \le 2$.

case 2 The algorithm returns from inside the for loop: hence there exists and index i' such that $sum + s_{i'} > t$. Observe that

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

and hence

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

that is

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

Exercise 2 (9 points) Let $X_1, X_2, \ldots, X_n$ be independent indicator random variables such that $Pr(X_i = 1) = 1/(4e)$. Let $X = \sum_{i=1}^n X_i$ and $\mu = E[X]$. By applying the following Chernoff bound, which holds for every $\delta > 0$,

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

prove that

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

Solution: To apply the Chernoff bound we set $n/2$ equal to $(1+\delta)\mu$; since $\mu = E[X] = \sum_{i=1}^n E[X_i] = n/(4e)$, we get $\delta = 2e - 1$. Therefore

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