

Cognome.....

# Algoritmi e Strutture Dati

Nome.....

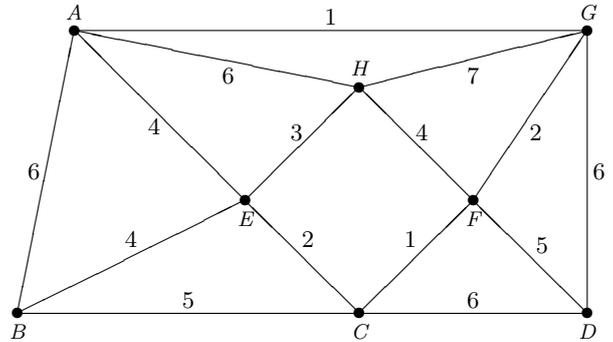
Prova scritta del 12 luglio 2019

Matricola.....

TEMPO DISPONIBILE: 2 ore

Le risposte agli esercizi 1 e 2 devono essere scritte negli appositi riquadri su questo foglio (risposte scritte su altri fogli non saranno considerate). La soluzione dell'esercizio 3 va scritta su uno dei fogli di protocollo forniti. Le brutte copie NON devono essere consegnate. Ricordatevi di scrivere cognome e nome su TUTTI i fogli (inclusi quelli di brutta).

1. Nei punti (a) e (b) seguenti, determinate un albero di ricoprimento minimo del grafo pesato rappresentato nella figura a destra, applicando l'algoritmo indicato. Scrivete l'elenco degli archi selezionati, in un ordine con cui sono scelti dall'algoritmo.



(a) Algoritmo di Kruskal.
(b) Algoritmo di Prim, iniziando dal vertice A.
(c) Scrivete la definizione di albero (in termini di grafi).

2. La seguente sequenza di numeri, memorizzata in un array, deve essere ordinata in modo crescente:  
12 46 17 10 14 11 35 13

(a) Supponete di ordinare la sequenza mediante l'algoritmo <b>radixSort</b> (rispetto alla base 10). Indicate il contenuto dell'array dopo la prima iterazione del ciclo principale dell'algoritmo.
(b) Supponete di ordinare la sequenza mediante l'algoritmo <b>quickSort</b> , scegliendo come perno il primo elemento. Indicate il contenuto dell'array dopo avere effettuato la partizione, prima delle chiamate ricorsive di <b>quickSort</b> .
(c) Supponete di ordinare la sequenza mediante l'algoritmo <b>insertionSort</b> . Indicate il contenuto dell'array dopo la terza iterazione del ciclo principale dell'algoritmo.
(d) Supponete di ordinare la sequenza mediante l'algoritmo <b>heapSort</b> . Indicate il contenuto dell'array <i>immediatamente prima</i> di effettuare lo scambio che colloca il numero 17 nella sua posizione finale all'interno dell'array.

**3.** Si vuole costruire un algoritmo che ricevendo due stringhe di caratteri  $s$  e  $t$  determini la *più lunga sottostringa comune* a  $s$  e  $t$  (o una qualunque sottostringa più lunga comune, nel caso ve ne sia più di una). Si ricorda che:

- una *sottostringa* di una stringa è una sequenza di caratteri *contigui* presi dalla stringa (la stringa vuota, formata da 0 caratteri, è una sottostringa di qualunque stringa);
- una sottostringa che comprende *tutti i caratteri* della stringa data *fino a una certa posizione* viene chiamata *prefisso*;
- una sottostringa che comprende *tutti i caratteri* della stringa data *a partire da una certa posizione* viene chiamata *suffisso*.

Ad esempio, la più lunga sottostringa comune a **camicia** e **amicizia** è **amici**. Le stringhe **camicia** e **sciama** hanno due sottostringhe comuni di lunghezza massima: **ami** e **cia**.

*Cosa si richiede.*

Sono date due stringhe  $s$  e  $t$  formate rispettivamente da  $n$  e da  $m$  caratteri, indicati con  $s_1, s_2, \dots, s_n$  e  $t_1, t_2, \dots, t_m$ . Per  $i = 0, \dots, n$ ,  $j = 0, \dots, m$ , sia  $c_{i,j}$  la lunghezza del più lungo suffisso comune delle stringhe  $s_1 s_2 \dots s_i$  e  $t_1 t_2 \dots t_j$ , cioè dei prefissi di lunghezza  $i$  e  $j$  delle stringhe  $s$  e  $t$ , rispettivamente. Ad esempio per  $s = \text{camicia}$  e  $t = \text{amicizia}$  si ha  $c_{4,3} = 3$ , e  $c_{4,5} = 1$  (la matrice completa  $C$  dei valori  $c_{i,j}$  è riportata alla fine dell'esercizio).

Risolvete i seguenti punti *nell'ordine indicato*.

- Scrivete una o più formule che permettano di ottenere il generico valore  $c_{i,j}$   $i = 1, \dots, n$ ,  $j = 1, \dots, m$ , in base a opportuni  $c_{i',j'}$  con  $i' < i$  o  $j' < j$  e ai caratteri  $s_i$  e  $t_j$  delle stringhe. Indicate anche il valore di  $c_{i,j}$  quando  $i$  o  $j$  valgono 0.
- Descrivete *sinteticamente a parole* e poi *ad alto livello* in pseudocodice, un algoritmo basato sulla tecnica di *programmazione dinamica* che ricevendo in ingresso le stringhe  $s$  e  $t$  calcoli la matrice  $C$  dei valori  $c_{i,j}$  (utilizzate quanto scritto al punto (a)) e stabilisca la lunghezza della più lunga sottostringa comune.
- Fornite una stima del tempo totale utilizzato dall'algoritmo in funzione di  $n$  e di  $m$ .
- Descrivete sinteticamente a parole come ricavare dalla matrice  $C$  la più lunga sottostringa comune (o una qualunque più lunga sottostringa comune, nel caso ve ne sia più di una).
- Fornite una stima del tempo utilizzato dalla parte di algoritmo descritta al punto (d) in funzione di  $n$  e di  $m$ .

**Importante.** Non cambiate i nomi stabiliti nel testo dell'esercizio. Indicate i caratteri delle stringhe direttamente come  $s_i$  o  $t_j$ , come nel testo dell'esercizio. Le risposte devono essere adeguatamente giustificate.

*Esempi.*

Nelle figure seguenti sono riportati esempi di matrice  $C$  per  $s = \text{camicia}$  e  $t = \text{amicizia}$ , e  $s = \text{camicia}$  e  $t = \text{sciama}$ . Per comodità sopra e a sinistra di ogni matrice sono riportati gli indici della matrice e i caratteri delle stringhe.

		0	1	2	3	4	5	6	7	8
			a	m	i	c	i	z	i	a
0		0	0	0	0	0	0	0	0	0
1	c	0	0	0	0	<b>1</b>	0	0	0	0
2	a	0	<b>1</b>	0	0	0	0	0	0	<b>1</b>
3	m	0	0	<b>2</b>	0	0	0	0	0	0
4	i	0	0	0	<b>3</b>	0	<b>1</b>	0	<b>1</b>	0
5	c	0	0	0	0	<b>4</b>	0	0	0	0
6	i	0	0	0	<b>1</b>	0	<b>5</b>	0	<b>1</b>	0
7	a	0	<b>1</b>	0	0	0	0	0	0	<b>2</b>

		0	1	2	3	4	5	6
			s	c	i	a	m	i
0		0	0	0	0	0	0	0
1	c	0	0	<b>1</b>	0	0	0	0
2	a	0	0	0	0	<b>1</b>	0	0
3	m	0	0	0	0	0	<b>2</b>	0
4	i	0	0	0	<b>1</b>	0	0	<b>3</b>
5	c	0	0	<b>1</b>	0	0	0	0
6	i	0	0	0	<b>2</b>	0	0	<b>1</b>
7	a	0	0	0	0	<b>3</b>	0	0