Algoritmi e Strutture Dati

a.a. 2020/21

Compito del 30/06/2021

Cognome: ___________________________ Nome: ___________________________

Matricola: ___________________________ E-mail: ___________________________

Parte I

(30 minuti; ogni esercizio vale 2 punti)

Avvertenza: Si giustifichino tecnicamente tutte le risposte. In caso di discussioni poco formali o approssimative gli esercizi non verranno valutati pienamente.

1. Si consideri una tabella Hash di dimensione $m = 8$, e indirizzamento aperto con doppio Hashing basato sulle funzioni $h_1(k) = k \bmod m$ e $h_2(k) = 1 + 2 * (k \bmod (m - 3))$. Si descriva in dettaglio come avviene l'inserimento della sequenza di chiavi: *25, 41, 68, 19*.

2. Un algoritmo ricorsivo MyMST è in grado di determinare un albero di copertura minimo all'interno di un grafo pesato G. La sua complessità è pari a:

$$T(m) = 4T(m/2) + 7(m^2 + 1) + 5\log m$$

   dove m rappresenta il numero di archi del grafo. Esistono algoritmi più efficienti di MyMST per risolvere il problema dato?

3. Siano $\mathcal{P}$ e $\mathcal{Q}$ due problemi in NP e si supponga $\mathcal{P} \leq_P \mathcal{Q}$. Si stabilisca se le seguenti affermazioni sono vere o false:

   (a) Se $\mathcal{Q}$ è risolvibile in tempo quadratico, allora $\mathcal{P}$ è risolvibile in tempo quadratico

   (b) Se $\mathcal{P}$ è risolvibile in tempo polinomiale, allora $\mathcal{Q}$ è risolvibile in tempo polinomiale

   (c) Se $\mathcal{P}$ è un problema NP-completo, allora $\mathcal{Q}$ è NP-completo

Algoritmi e Strutture Dati

a.a. 2020/21

Compito del 30/06/2021

Cognome: ______________________________ Nome: ______________________________

Matricola: ______________________________ E-mail: ______________________________

Parte II

(2.5 ore; ogni esercizio vale 6 punti)

Avvertenza: Si giustifichino tecnicamente tutte le risposte. In caso di discussioni poco formali o approssimative gli esercizi non verranno valutati pienamente.

1. Dato un albero binario i cui nodi contengono interi, si vuole aggiungere ad ogni foglia un figlio contenente la somma dei valori che appaiono nel cammino dalla radice a tale foglia. Se la somma di tali valori è positiva sarà aggiunto come figlio sinistro, altrimenti come figlio destro.
   i) Scrivere una procedura **efficiente in C** *aggiungiFigli(Node u)* che dato in input la radice *u* di un albero *T* aggiunge alle foglie i figli come sopra specificato.
   ii) Valutarne la complessità, **indicando eventuali relazioni di ricorrenza.**
   iii) Scrivere il tipo *Node* in C.

2. Sia A un array di lunghezza $n - k$ con $k \geq 2$, privo di ripetizioni e contenente interi nell'intervallo $[n^2 + 1, n^2 + n]$. Si consideri il problema di determinare i k numeri interi appartenenti all'intervallo $[n^2 + 1, n^2 + n]$ che non compaiono in A.
   Si scriva una procedura **efficiente** che, dati A, n e k, risolva il problema proposto stampando gli interi che non compaiono in A. Calcolarne la complessità.

3. Sia $G = (V, E)$ un grafo orientato e pesato, con funzione peso $w : E \to \mathrm{R}$, che non contenga cicli negativi e cappi. Si assuma che i vertici siano numerati da 1 a n, ovvero $V = \{1, \dots, n\}$, e sia W la matrice di dimensione n x n definita come segue:

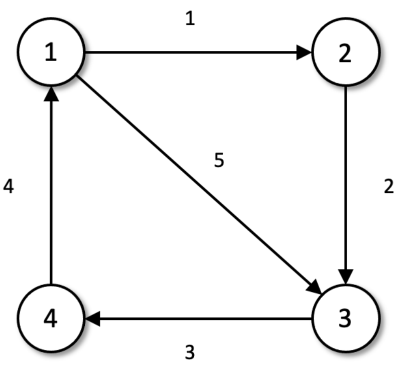
$$W[i,j] = \begin{cases} 0 & se\ \ i = j \\ w(i,j) & se\ \ i \neq i\ \ e\ \ (i,j) \in E \\ +\infty & se\ \ i \neq j\ \ e\ \ (i,j) \notin E \end{cases}$$

   Si stabilisca quali problemi risolvono i due algoritmi riportati di seguito (che prendono in ingresso la matrice W associata al grafo), se ne dimostri la correttezza e se ne calcoli la complessità.

```
MyAlgorithm1(W)
n = n.ro di righe di W
alloca spazio per un vettore D
di dimensione n
D[1] = 0
for i = 2 to n
  D[i] = +∞
for k = 1 to n
  for i = 1 to n
    for j = 1 to n
      D[j] = min{ D[j], D[i] + W[i,j] }
return D
```

```
MyAlgorithm2(W)
n = n.ro di righe di W
alloca spazio per una matrice D
di dimensione n x n
for i = 1 to n
  for j = 1 to n
    D[i,j] = W[i,j]
for k = 1 to n
  for i = 1 to n
    for j = 1 to n
      D[i,j] = min{ D[i,j], D[i,k] + D[k,j] }
return D
```

   Cosa restituiscono in uscita i due algoritmi in presenza del grafo seguente? (Scrivere <u>esplicitamente</u> il *vettore D* nel primo caso e la *matrice D* nel secondo. Non è necessario riportare la simulazione degli algoritmi.)



4. La seguente tabella fornisce le distanze (in unità di 100 miglia) tra gli aeroporti delle città di Londra, Città del Messico, New York, Parigi, Pechino e Tokyo:

	L	*CM*	*NY*	*Pa*	*Pe*	*T*
L	–	56	35	2	50	60
CM	56	–	21	57	78	70
NY	35	21	–	36	68	67
Pa	2	57	36	–	51	61
Pe	50	78	68	51	–	13
T	60	70	67	61	13	–

Utilizzando l'algoritmo di Prim, si determini un albero di copertura minimo per il grafo corrispondente.

In particolare, utilizzando il vertice *L* come sorgente:

a) si indichi l'ordine con cui vengono estratti i vertici
b) si riempia la tabella seguente, con i valori dei campi key e π, per ogni singola iterazione

È possibile, utilizzando un altro vertice sorgente, determinare un albero di copertura minimo diverso da quello trovato? In caso affermativo lo si determini (nel qual caso, non è necessario riportare la simulazione dettagliata dell'algoritmo), altrimenti si fornisca una motivazione formale.

	vertice *L*		**vertice *CM***		**vertice *NY***		**vertice *Pa***		**vertice *Pe***		**vertice *T***	
	key	**π**	**key**	**π**	**key**	**π**	**key**	**π**	**key**	**π**	**key**	**π**
dopo inizializzazione												
iterazione 1												
iterazione 2												
iterazione 3												
iterazione 4												
iterazione 5												
iterazione 6												