

Algoritmi e Strutture Dati

a.a. 2023/24

Compito del 9/9/2024

Cognome: _____

Nome: _____

Matricola: _____

E-mail: _____

Parte I

(30 minuti; ogni esercizio vale 2 punti)

Avvertenza: Si giustificino 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 $m = 13$, e indirizzamento aperto con doppio Hashing basato sulle funzioni $h_1(k) = k \bmod m$ e $h_2(k) = 1 + (k \bmod (m - 2))$. Si descriva in dettaglio come avviene l'inserimento della sequenza di chiavi: 42, 16, 7, 85.

2. Si stabilisca se la seguente affermazione è vera o falsa:

“Se $G = (V, E)$ è un grafo non orientato con k componenti connesse, allora $|E| \geq |V| - k$.”

In caso affermativo si fornisca una dimostrazione, altrimenti un controesempio.

3. Un algoritmo per determinare i cammini minimi tra tutte le coppie di vertici di un grafo pesato ha complessità

$$T(n) = 9T\left(\frac{n}{2}\right) + 9n^2 + 2 \log n$$

dove n rappresenta il numero di vertici del grafo in ingresso. Si dica se l'algoritmo in questione è preferibile o meno all'algoritmo di Floyd-Warshall e perché.

Algoritmi e Strutture Dati

a.a. 2023/24

Compito del 9/9/2024

Cognome: _____

Nome: _____

Matricola: _____

E-mail: _____

Parte II

(2.5 ore; ogni esercizio vale 6 punti)

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

1. Siano T_1 e T_2 due alberi binari completi aventi lo stesso numero di nodi che contengono interi e con i seguenti campi: **key**, **left** e **right**.

- a. Si scriva una procedura **efficiente** che vada a sottrarre ai nodi di T_1 i valori dei nodi di T_2 . Per esempio:

$$\begin{array}{cccccc} & 1 & & & 0 & & & 1 \\ 3 & 4 & & 7 & 10 & & -4 & -6 \\ 5 & 6 & 7 & 8 & 2 & 3 & 5 & 4 & 3 & 3 & 2 & 4 \end{array} =$$

Il prototipo della procedura è:

```
void sottrazione_alberi(PNode r1, PNode r2)
```

dove $r1$ è la radice di T_1 e $r2$ è la radice di T_2 .

- b. Valutare e giustificare la complessità della procedura, **indicando eventuali relazioni di ricorrenza e mostrando la loro risoluzione**.
 - c. Specificare il linguaggio di programmazione scelto e la **definizione** di PNode.
2. Sia A un vettore di n interi e si consideri il problema di determinare se esistono due interi che occorrono in A lo stesso numero di volte.
 - a. Si scriva un algoritmo **efficiente** per il problema proposto.
 - b. Si scriva un algoritmo **efficiente** per il problema proposto nel caso in cui in A occorrono **c valori distinti**, dove **c** è una **costante intera positiva**.
 - c. Si determini e giustifichi la complessità degli algoritmi proposti.

Si devono scrivere eventuali funzioni/procedure ausiliarie utilizzate.

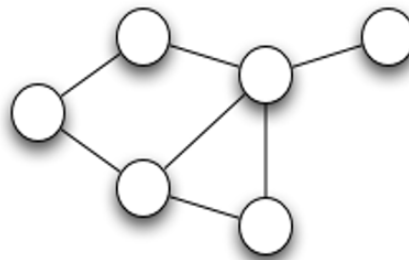
3. Si stabilisca quale problema risolve il seguente algoritmo, dove A rappresenta la matrice di adiacenza di un grafo non orientato (si assuma che i vertici siano numerati da 1 a n):

```
MyAlgorithm(A)
1. n = numero di righe di A          /* determina il n.ro di vertici del grafo */
2. S = ∅
3. for i = 1 to n
4.   if MyFunction(S,i) then
5.     S = S ∪ {i}
6. return S

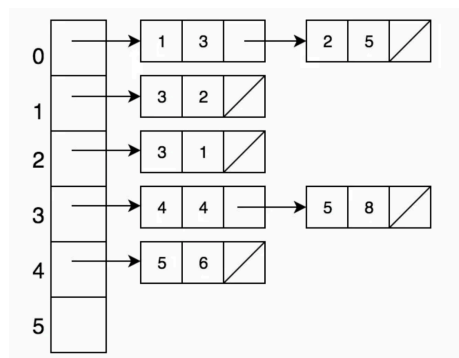
MyFunction(S,i)
1. for each j ∈ S
2.   if a[i,j] = 1 then
3.     return FALSE
4. return TRUE
```

Si dimostri la correttezza dell'algoritmo e si determini la sua complessità.

Si simuli infine accuratamente la sua esecuzione sul seguente grafo e si mostri (con alcuni esempi) che il risultato finale può dipendere dal modo in cui vengono numerati i vertici del grafo. In particolare, si determini una numerazione che consentirebbe all'algoritmo di restituire in uscita il massimo numero di vertici.



4. Si scriva l'algoritmo di Dijkstra, si dimostri la sua correttezza, si fornisca la sua complessità computazionale e si simuli accuratamente la sua esecuzione sul seguente grafo rappresentato mediante lista di adiacenza:



In particolare:

- come sorgenti, si utilizzi sia il vertice 0 che il vertice 5
- in entrambi i casi si indichi l'ordine con cui vengono estratti i vertici
- si riempiano le tabelle seguenti con i valori dei vettori d e π , iterazione per iterazione

Vertice sorgente: 0

[illegible]

Vertice sorgente: 5[illegible]