Iniziato	lunedì, 12 giugno 2023, 18:00
Stato	Completato
Terminato	lunedì, 12 giugno 2023, 18:30
Tempo impiegato	30 min. 17 secondi
Valutazione	**7,56** su un massimo di 8,10 (**93,33**%)

Domanda 1

Risposta corretta

Punteggio ottenuto 0,27 su 0,27

Le tabelle ad accesso diretto...

Scegli un'alternativa:

a. sono estremamente vantaggiose in termini di occupazione dello spazio: se nel dizionario sono presenti n elementi, occupo uno spazio proporzionale ad n. In particolare, se il dizionario è vuoto non occupo spazio.

b. sono molto rigide: le chiavi possono solo essere dei numeri interi, ed il loro range deve essere per forza [0, m-1], con m dimensione della tabella ✔

c. sono estremamente flessibili: le chiavi possono essere elementi qualunque, ad esempio stringhe, caratteri, etc.

La risposta corretta è: sono molto rigide: le chiavi possono solo essere dei numeri interi, ed il loro range deve essere per forza [0, m-1], con m dimensione della tabella

Domanda 2

Risposta corretta

Punteggio ottenuto 0,27 su 0,27

Si consideri la struttura dati "liste di adiacenza con vertici memorizzati in un array" per rappresentare un grafo non orientato con **n** nodi ed **m** archi. La complessità di **incidentEdges** chiamata sul vertice **v** è

Scegli un'alternativa:

a. Theta(1) sia nel caso migliore che nel caso peggiore

b. Theta(1) nel caso migliore e Theta (n) nel caso peggiore

c. Theta (grado (v)) sia nel caso migliore che nel caso peggiore ✔

d. Theta(1) nel caso migliore e Theta (grado (v)) nel caso peggiore

La risposta corretta è: Theta (grado (v)) sia nel caso migliore che nel caso peggiore

Domanda 3

Risposta corretta

Punteggio ottenuto 0,27 su 0,27

Durante il corso abbiamo accennato a un algoritmo per cercare i cammini minimi in un grafo con pesi non negativi sugli archi. Come si chiama questo algoritmo?

Scegli un'alternativa:

a. Algoritmo di ricerca in ampiezza (BFS)

b. Algoritmo di Dijkstra ✔

c. Algoritmo di ricerca in profondità (DFS)

d. Algoritmo degli alberi AVL

La risposta corretta è: Algoritmo di Dijkstra

Domanda 4

Risposta corretta

Punteggio ottenuto 0,27 su 0,27

Quale delle seguenti affermazioni sui Binary Search Tree (BST) è corretta

Scegli un'alternativa:

a. Un BST è un **albero binario** con alcune proprietà che le chiavi associate ai nodi devono rispettare ✔

b. Un BST è un **albero binario completo** con alcune proprietà che le chiavi associate ai nodi devono rispettare

c. Un BST è un **albero binario quasi completo** con alcune proprietà che le chiavi associate ai nodi devono rispettare

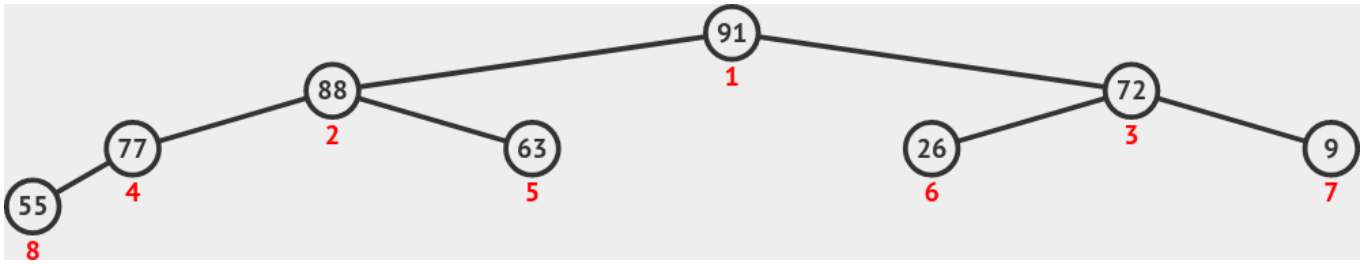
La risposta corretta è: Un BST è un **albero binario** con alcune proprietà che le chiavi associate ai nodi devono rispettare

Domanda 5

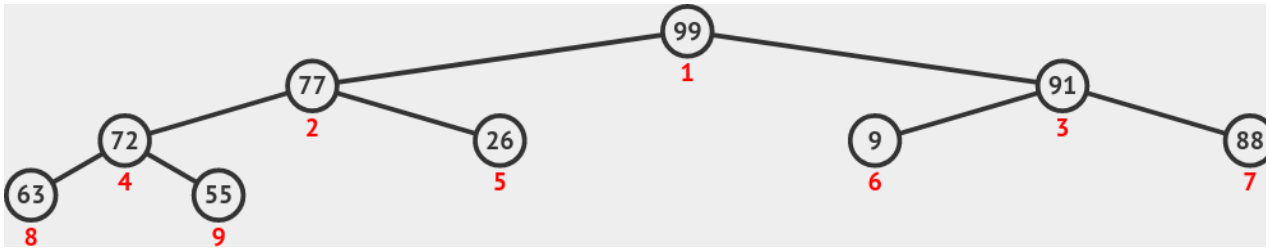
Risposta corretta

Punteggio ottenuto 0,27 su 0,27

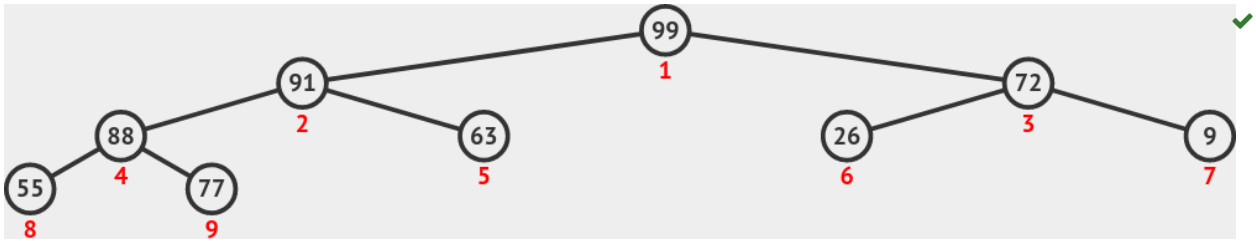
Si consideri questo heap binario di tipo max. Si ignorino i numeri scritti in rosso (indicano la cella dell'array in cui la etichetta del nodo è contenuta e non vi servono). Come viene modificato lo heap a seguito dell'inserimento del numero **99**?



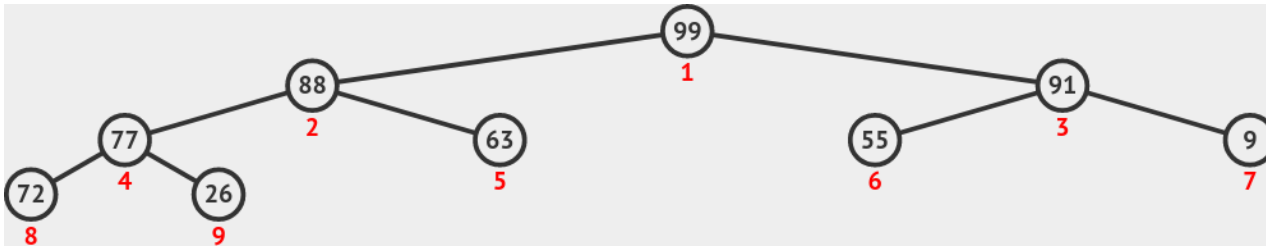
a.



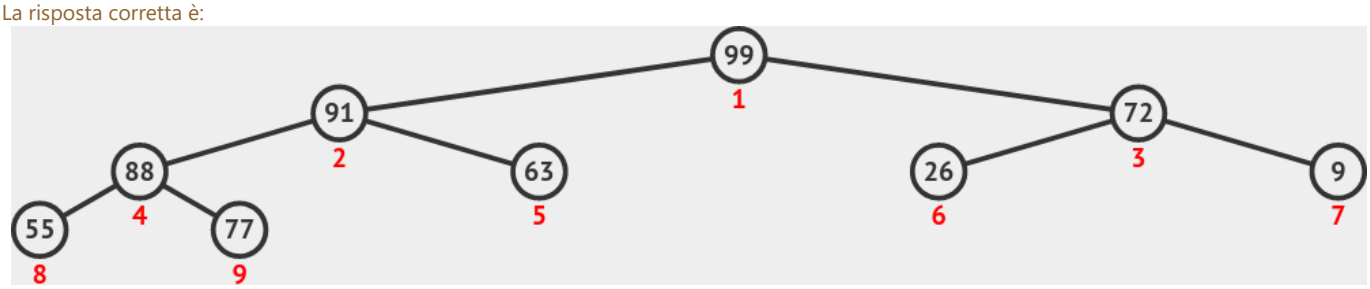
b. ✔



c.



La risposta corretta è:



Domanda 6

Risposta corretta

Punteggio ottenuto 0,27 su 0,27

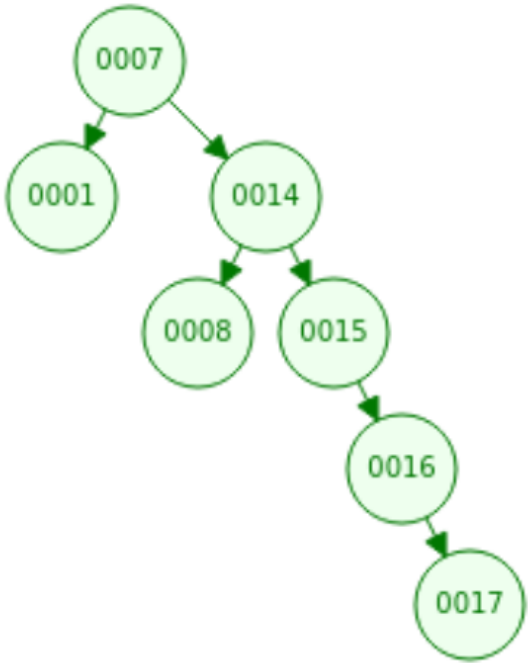
Si consideri un Binary Search Tree inizialmente vuoto, sul quale si eseguono le seguenti operazioni di inserimento (nelle quali come al solito mostriamo solo la chiave), nell'ordine in cui le vedete dall'alto in basso: si inserisce prima l'elemento con chiave 7, poi l'elemento con chiave 14, etc.

```
insert(7, ....);
insert(14, ....);
insert(8, ....);
insert(15, ....);
insert(16, ....);
insert(17, ....);
insert(1, ....);
```

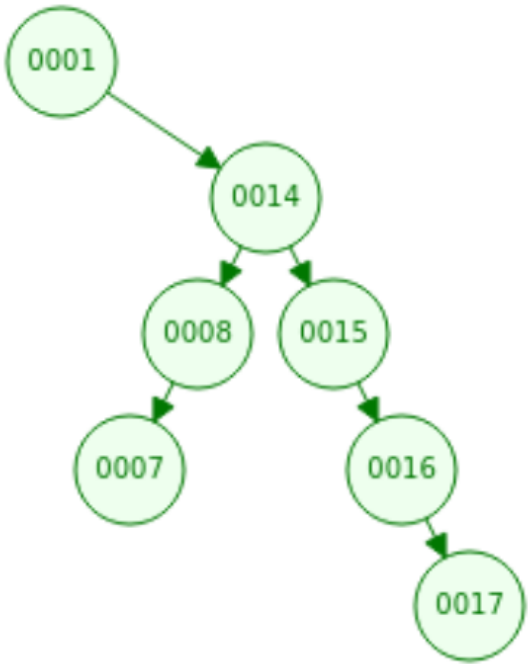
Qual è il BST risultante dagli inserimenti?

Scegli un'alternativa:

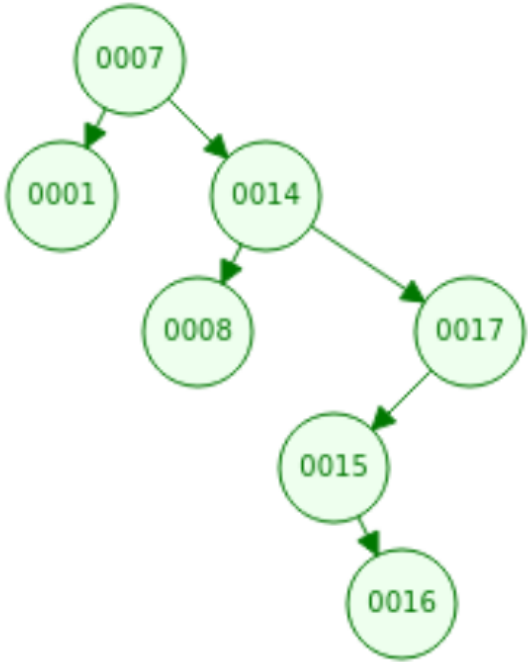
a. ✔



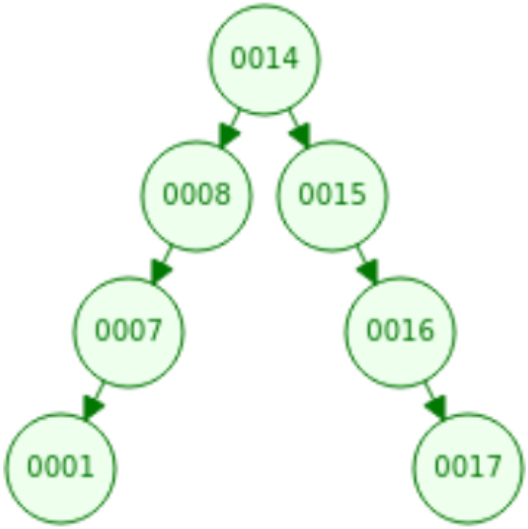
b.

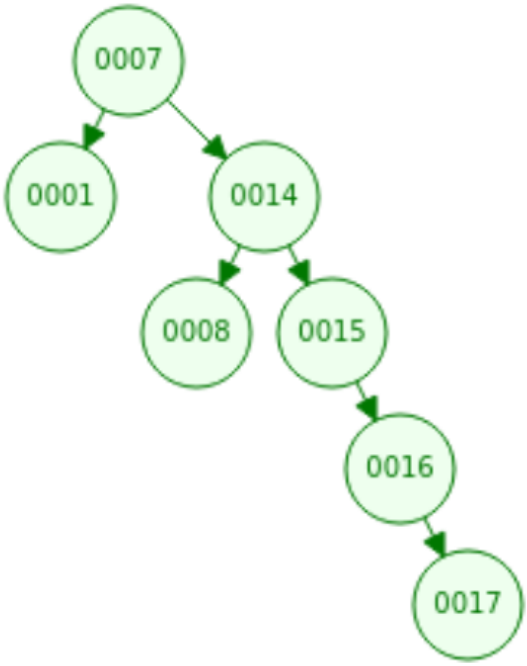


c.



d.





La risposta corretta è:

Domanda 7

Risposta corretta

Punteggio ottenuto 0,27 su 0,27

In quale situazione la complessità temporale di quicksort si calcola seguendo un approccio simile a quello del calcolo della complessità temporale di mergesort?

Scegli un'alternativa:

- a. Quando come pivot viene sempre selezionato il **mediano** degli elementi su cui quicksort viene chiamata ✔
- b. Quando come pivot viene sempre selezionato il valore che si ottiene come **media aritmetica** degli elementi su cui quicksort viene chiamata
- c. Quando come pivot viene sempre selezionato **il minimo o il massimo** degli elementi su cui quicksort viene chiamata

La risposta corretta è: Quando come pivot viene sempre selezionato il **mediano** degli elementi su cui quicksort viene chiamata

Domanda 8

Risposta corretta

Punteggio ottenuto 0,27 su 0,27

Contesto: Caso peggiore di lettura di **n** dati da uno stream e inserimento in **ultima posizione** in lista collegata semplicemente, **con controllo che nella lista non ci siano elementi duplicati.**

Si consideri il seguente codice:

```
void read_list(istream& input_stream, basic_list& list)
{
  create_empty(list);
  int d;
  while (ReadData(input_stream, d))
      last_insert_no_duplicates(list, d);
}
```

Qual è la complessità temporale della funzione *read_list* nel **caso peggiore?**

Si supponga che gli elementi letti da *input_stream* siano **n** e che *list* sia una lista di interi semplice (collegata semplicemente, non circolare, senza sentinella).

create_empty inizializza list come lista vuota e ha complessità costante nel caso migliore e peggiore; R*eadData* legge un elemento alla volta da *input_stream* restituendo false se non ci sono più elementi da leggere e true altrimenti; ha complessità costante nel caso migliore e peggiore.

last_insert_no_duplicates inserisce un elemento **nell'ultima posizione di *list*** solo se l'elemento **non è già presente in list**; se è presente, non lo re-inserisce.

Scegli un'alternativa:

- a. Theta (n !)
- b. Theta (n)
- c. Theta (n^2) ✔
- d. Theta (log n)
- e. Theta (2^n)
- f. Theta (1)

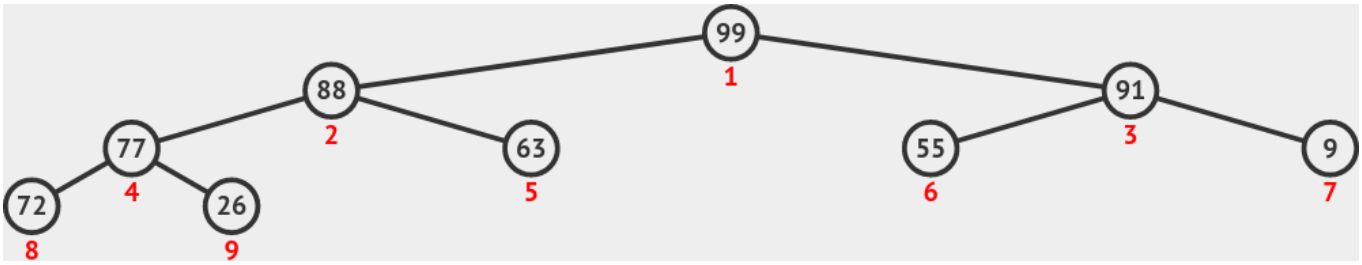
La risposta corretta è: Theta (n^2)

Domanda 9

Risposta corretta

Punteggio ottenuto 0,27 su 0,27

Si consideri l'albero binario radicato in "99" rappresentato sotto (si ignorino i numeri in rosso): qual è l'ordine con cui vengono visitati i nodi con una visita simmetrica?



- a. 72-26-77-63-55-9-88-91-99
- b. 72-77-26-88-63-99-55-91-9 ✔
- c. 99-88-91-77-63-55-9-72-26
- d. 9-26-55-63-72-77-88-91-99

La risposta corretta è:
72-77-26-88-63-99-55-91-9

Domanda 10

Risposta corretta

Punteggio ottenuto 0,27 su 0,27

Un albero **d-ario** è un albero in cui....

Scegli un'alternativa:

- a. tutti i nodi **tranne la radice** hanno **esattamente grado d**
- b. i nodi hanno **al più grado d** ✔
- c. tutti i nodi **tranne le foglie** hanno **esattamente grado d**

La risposta corretta è: i nodi hanno **al più grado d**

Domanda 11

Risposta corretta

Punteggio ottenuto 0,27 su 0,27

Contesto:
Analisi di complessità delle operazioni su liste, Struttura dati: array dinamico con size e maxsize (e ridimensionamento)

Dimensione della lista: n

La posizione di Elem appartiene a N, insieme dei numeri naturali

Nel contesto indicato sopra, qual è la complessità temporale, nel caso migliore e peggiore, di **set: N x Elem x List → List**

Attenzione: N non è la dimensione della lista ma è l'insieme dei numeri naturali, da cui si prende un valore che indica la posizione di Elem

Scegli un'alternativa:

- a. Theta(1) nel caso migliore, Theta(n) nel caso peggiore
- b. Theta(1) nel caso migliore e peggiore ✔
- c. Theta(n) nel caso migliore e peggiore

La risposta corretta è: Theta(1) nel caso migliore e peggiore

Domanda 12

Risposta corretta

Punteggio ottenuto 0,27 su 0,27

Sia h (n) = n^2 + 2 n! + 2 n log n + 2 n + 2^n + 3 n!
Selezionare l'unica risposta corretta.

Scegli un'alternativa:

- a. h (n) appartiene a Theta (n log n)
- b. h (n) appartiene a O (1)
- c. h (n) appartiene a O (n !) ✔
- d. h (n) appartiene a Theta (2^n)

La risposta corretta è: h (n) appartiene a O (n !)

Domanda 13

Risposta corretta

Punteggio ottenuto 0,27 su 0,27

Un cammino in un grafo è

Scegli un'alternativa:

a. una sequenza di vertici $[v_0, v_1, \ldots v_0]$ tali che ogni coppia di vertici consecutivi nella sequenza (v_i, v_{i+1}) è connessa da un arco e il primo vertice coincide con l'ultimo (si noti che v_0 è il primo e l'ultimo elemento della sequenza)

b. una sequenza di vertici $[v_0, v_1, \ldots v_n]$ tali che ogni coppia di vertici consecutivi nella sequenza (v_i, v_{i+1}) è connessa da un arco ✔

c. una sequenza di vertici $[v_0, v_1, \ldots v_n]$ tali che ogni coppia di vertici consecutivi nella sequenza (v_i, v_{i+1}) è connessa da un arco e non ci sono vertici ripetuti nella sequenza

La risposta corretta è: una sequenza di vertici $[v_0, v_1, \ldots v_n]$ tali che ogni coppia di vertici consecutivi nella sequenza (v_i, v_{i+1}) è connessa da un arco

Domanda 14

Risposta corretta

Punteggio ottenuto 0,27 su 0,27

Si consideri la struttura dati "liste di adiacenza con vertici memorizzati in una lista" per rappresentare un grafo non orientato con **n** nodi ed **m** archi. La complessità temporale di **removeVertex** del vertice **v** è, **nel caso peggiore**

Scegli un'alternativa:

a. Theta(grado(v))

b. Theta(n + m) ✔

c. Theta (m)

d. Theta(n)

La risposta corretta è: Theta(n + m)

Domanda 15

Risposta corretta

Punteggio ottenuto 0,27 su 0,27

Siano **m** = dimensione della tabella di hash ed **n** = numero di elementi presenti nel dizionario; come si definisce il fattore di carico **alpha**?

Scegli un'alternativa:

a. alpha = m/n

b. alpha = v[h(k)]

c. alpha = n/m ✔

La risposta corretta è: alpha = n/m

Domanda 16

Risposta corretta

Punteggio ottenuto 0,27 su 0,27

Si consideri la struttura dati "matrice di adiacenza" per rappresentare un grafo non orientato con **n** nodi ed **m** archi. La complessità temporale di **addEdge** dell'arco **(x,y)** è, **nel caso peggiore**

Scegli un'alternativa:

a. Theta (n^2) perché potrei dover ridimensionare la matrice

b. Theta (m)

c. Theta (n)

d. Theta (minimo(grado (x), grado (y)))

e. Theta(1) ✔

La risposta corretta è: Theta(1)

Domanda 17

Risposta corretta

Punteggio ottenuto 0,27 su 0,27

I Binary Search Tree e gli heap binari

Scegli un'alternativa:

a. sono entrambi alberi binari completi

b. sono entrambi alberi generici in cui l'altezza può variare da n (con n numero dei nodi) a n^2

c. sono entrambi alberi generici in cui l'altezza è sempre al più log n, ma può essere anche minore

d. sono entrambi alberi binari ✔

e. sono entrambi alberi binari quasi completi

La risposta corretta è: sono entrambi alberi binari

Domanda 18

Risposta corretta

Punteggio ottenuto 0,27 su 0,27

Contesto:
Analisi di complessità delle operazioni su set, struttura dati: array **ORDINATO** con size e maxsize.

Dimensione del primo insieme: **n**
Dimensione del secondo insieme: **m**

Nel contesto indicato sopra, qual è la complessità temporale nel caso peggiore di **setUnion: Set x Set → Set**

Scegli un'alternativa:

- a. Theta (n + m) ✔
- b. Theta (n * m)
- c. Theta (n)
- d. Theta (m)

La risposta corretta è: Theta (n + m)

Domanda 19

Risposta corretta

Punteggio ottenuto 0,27 su 0,27

Si consideri il **TDD Coda con Priorità** implementato con uno **heap binario** con **n** nodi.

Qual è la complessità temporale dell'operazione **deleteMax: PriorityQueue -> PriorityQueue** nel caso peggiore?

Scegli un'alternativa:

- a. Theta (1)
- b. Theta (n)
- c. Theta (log n) ✔

La risposta corretta è: Theta (log n)

Domanda 20

Risposta corretta

Punteggio ottenuto 0,27 su 0,27

Un grafo **G'** è detto albero di ricoprimento di un grafo **G** se

Scegli un'alternativa:

- a. **G'** è un sottografo di ricoprimento di **G** ed è anche un albero libero ✔
- b. **G'** ha gli stessi archi di **G**, ha un sottoinsieme proprio dei vertici di **G** ed è privo di cicli
- c. **G'** ha gli stessi vertici di **G** ed è un grafo completo

La risposta corretta è: **G'** è un sottografo di ricoprimento di **G** ed è anche un albero libero

Domanda 21

Risposta corretta

Punteggio ottenuto 0,27 su 0,27

Sia h (n) = n^3 + 6 n! + n log n + 6 n + 2^n
Selezionare l'unica risposta corretta.

Scegli un'alternativa:

- a. h (n) appartiene a O (n !) ✔
- b. h (n) appartiene a Theta (n log n)
- c. h (n) appartiene a O (1)
- d. h (n) appartiene a Theta (2^n)

La risposta corretta è: h (n) appartiene a O (n !)

Domanda 22

Risposta corretta

Punteggio ottenuto 0,27 su 0,27

Una struttura dati indicizzata (cioè un array) supporta l'implementazione efficiente delle operazioni del **TDD Tree** quando

Scegli un'alternativa:

- a. l'albero da rappresentare è un BST
- b. l'albero da rappresentare è generico
- c. l'albero da rappresentare è un albero d-ario quasi completo ✔

La risposta corretta è: l'albero da rappresentare è un albero d-ario quasi completo

Domanda 23

Risposta corretta

Punteggio ottenuto 0,27 su 0,27

Si consideri il **TDD Dictionary** implementato con un **Binary Search Tree** con **n** nodi.

Qual è la complessità temporale dell'operazione **deleteElem: Key x Dictionary -> Dictionary** nel caso peggiore?

Scegli un'alternativa:

- a. Theta (log n)
- b. Theta (1)
- c. Theta (h), con h altezza del BST ✔

La risposta corretta è: Theta (h), con h altezza del BST

Domanda 24

Risposta corretta

Punteggio ottenuto 0,54 su 0,54

Questa domanda vale 0,54 punti

Si consideri il seguente codice:

```
bool paperino(Label aaa, Label bbb, Weight w, Graph& g) {
  if (aaa == bbb)
    return false;
  if (!isVertexInGraph(aaa, g) || !isVertexInGraph(bbb, g))
    return false;
  if (isEdgeInGraph(aaa,bbb,g) || isEdgeInGraph(bbb,aaa,g))
    return false;
  addHalfEdge(aaa, bbb, w, g);
  addHalfEdge(bbb, aaa, w, g);
  return true;
}
```

Gli identificatori "paperino", "aaa", "bbb" sono offuscati; gli altri hanno il significato intuitivo associato al loro nome.

Di quale funzione si tratta?

a. Della funzione search in un grafo orientato e pesato implementato con liste di adiacenza: cerca il vertice la cui etichetta è "w" nel grafo "g". Restituisce "false" se non lo trova e "true" se lo trova

b. Della funzione deleteMax in un binary search tree rappresentato come albero radicato: la radice è rappresentata dal nodo "aaa" e la nuova radice dal nodo "bbb"

c. Della funzione removeVertex in un grafo orientato e pesato implementato con matrici di adiacenza: rimuove il vertice la cui etichetta è "bbb" dal grafo "g". Restituisce "false" se nessun nodo con etichetta "bbb" è presente nel grafo

d. Della funzione addEdge in un grafo pesato e non orientato implementato con liste di adiacenza: aggiunge un arco di peso "w" tra i nodi con etichetta "aaa" e "bbb" nel grafo "g". Restituisce "true" solo se l'aggiunta è possibile e ha successo. ✔

e. Della funzione search in uno heap binario rappresentato come grafo orientato: si cerca un nodo la cui etichetta è "w" ed è compresa tra "aaa" e "bbb"

f. Della funzione addVertex in un grafo orientato e non pesato implementato con liste di adiacenza: aggiunge un nuovo vertice la cui etichetta è "aaa" al grafo "g". Restituisce "false" se un nodo con etichetta "aaa" è gia' presente

La risposta corretta è:
Della funzione addEdge in un grafo pesato e non orientato implementato con liste di adiacenza: aggiunge un arco di peso "w" tra i nodi con etichetta "aaa" e "bbb" nel grafo "g". Restituisce "true" solo se l'aggiunta è possibile e ha successo.

Domanda 25

Risposta corretta

Punteggio ottenuto 0,27 su 0,27

Contesto: si consideri **merge**, funzione ausiliaria di **mergeSort.**

Sia A un array di lunghezza **n**; siano **inizio** e **fine** gli indici che **delimitano la porzione di A** (con inizio e fine compresi) sulla quale viene chiamata merge per fondere la **prima metà della porzione di A compresa tra inizio e fine** con la **seconda metà della porzione di A compresa tra inizio e fine**. Qual è la complessità temporale di merge nel caso migliore e peggiore?

Scegli un'alternativa:

- a. Theta (fine-inizio) nel caso migliore e peggiore ✔
- b. Theta (n) nel caso migliore e peggiore
- c. Theta (n) nel caso peggiore e Theta (1) nel caso migliore, quello in cui n = 1

La risposta corretta è: Theta (fine-inizio) nel caso migliore e peggiore

Domanda 26

Risposta corretta

Punteggio ottenuto 0,27 su 0,27

Nella struttura dati "tabella di hash con liste di collisione" una buona funzione di hash dovrebbe essere...

Scegli un'alternativa:

- a. calcolabile in tempo **lineare** nel numero di elementi nella tabella e **bigettiva**
- b. calcolabile in tempo **costante** e **bigettiva**
- c. calcolabile in tempo **costante** e **iniettiva**
- d. calcolabile in tempo **lineare** nel numero di elementi nella tabella e in grado di **distribuire in modo uniforme** le chiavi nello spazio degli indici della tabella
- e. calcolabile in tempo **costante** e **surgettiva**
- f. calcolabile in tempo **lineare** nel numero di elementi nella tabella e **iniettiva**
- g. calcolabile in tempo **costante** e in grado di **distribuire in modo uniforme** le chiavi nello spazio degli indici della tabella ✔

La risposta corretta è: calcolabile in tempo **costante** e in grado di **distribuire in modo uniforme** le chiavi nello spazio degli indici della tabella

Domanda 27

Risposta errata

Punteggio ottenuto 0,00 su 0,27

Formalmente, a cosa corrisponde **l'interfaccia di un tipo di dato**?

Scegli un'alternativa:

- a. a una segnatura many-sorted, ovvero eterogenea
- b. a un insieme
- c. a una struttura dati
- d. a un'algebra, solitamente eterogenea, su una segnatura ✖
- e. a un elemento di un insieme

La risposta corretta è: a una segnatura many-sorted, ovvero eterogenea

Domanda 28

Risposta errata

Punteggio ottenuto 0,00 su 0,27

Si consideri il seguente algoritmo di ordinamento che chiameremo "cosaSort":

```
void cosaSort(vector<int>& v)
{
   int current, prev;
   unsigned int size = v.size();
   for (unsigned int i=1; i<size; ++i)
   { current=i;
     prev=i-1;
     while(prev>=0 && v[current]<v[prev])
        {
         scambia(v, current, prev);
         --current;
         --prev;
        }
   }
}
```

Che algoritmo è, e qual è la sua complessità temporale nel caso migliore e peggiore?

Scegli un'alternativa:

- a. Si tratta di selectionSort che ha complessità temporale Theta (n log n) nel caso migliore e Theta (n^2) nel caso peggiore
- b. Si tratta di insertionSort che ha complessità temporale Theta (n) nel caso migliore e Theta (n^2) nel caso peggiore
- c. Si tratta di selectionSort che ha complessità temporale Theta (n) nel caso migliore e Theta (n^2) nel caso peggiore
- d. Si tratta di insertionSort che ha complessità temporale Theta (n log n) nel caso migliore e Theta (n^2) nel caso peggiore
- e. Si tratta di insertionSort che ha complessità temporale Theta (n^2) nel caso migliore e peggiore ✖
- f. Si tratta di selectionSort che ha complessità temporale Theta (n^2) nel caso migliore e peggiore

La risposta corretta è: Si tratta di insertionSort che ha complessità temporale Theta (n) nel caso migliore e Theta (n^2) nel caso peggiore

Domanda 29

Risposta corretta

Punteggio ottenuto 0,27 su 0,27

Indichiamo come sempre con **n** il numero dei nodi di un albero e con **h** la sua altezza. Gli alberi rosso-neri sono...

Scegli un'alternativa:

a. heap binari inventati da A. Romagnoli; sono heap binari normali ma con i nodi colorati: i nodi ai livelli pari sono rossi e i nodi ai livelli dispari sono neri.

b. binary search tree la cui altezza è sempre n, come il numero dei nodi

c. binary search tree in cui l'altezza è limitata in modo da garantire che la complessità delle operazioni nel caso peggiore sia in Theta (log n) invece che in Theta (h) ✔

La risposta corretta è: binary search tree in cui l'altezza è limitata in modo da garantire che la complessità delle operazioni nel caso peggiore sia in Theta (log n) invece che in Theta (h)