

**Iniziato** lunedì, 27 febbraio 2023, 15:52

**Stato** Completato

**Terminato** lunedì, 27 febbraio 2023, 15:53

**Tempo impiegato** 28 secondi

**Punteggio** 1,00/1,00

**Valutazione** 10,00 su un massimo di 10,00 (100%)

**Domanda 1**

Risposta corretta

Punteggio ottenuto 1,00 su 1,00

Il principio di induzione presenta forti analogie con...

Scegli un'alternativa:

- a. ...il principio secondo cui si dovrebbe ingegnerizzare il SW, con un caso base che è la compilazione, ed un passo induttivo che è l'esecuzione
- b. ...un approccio ricorsivo alla soluzione dei problemi: la ricorsione deve avere un caso base e una definizione induttiva in cui ✓ Vero si fa un atto di fede sull fatto che la funzione dia il risultato atteso su una struttura dati più piccola, e si usa questa assunzione per programmare il comportamento sulla struttura dati più grossa
- c. ...il meccanismo con cui la memoria di un computer è organizzata, con uno stack analogo al caso base ed uno heap analogo al passo induttivo

Risposta corretta.

La risposta corretta è: ...un approccio ricorsivo alla soluzione dei problemi: la ricorsione deve avere un caso base e una definizione induttiva in cui si fa un atto di fede sull fatto che la funzione dia il risultato atteso su una struttura dati più piccola, e si usa questa assunzione per programmare il comportamento sulla struttura dati più grossa

**Iniziato** lunedì, 6 marzo 2023, 20:36

**Stato** Completato

**Terminato** lunedì, 6 marzo 2023, 20:37

**Tempo impiegato** 32 secondi

**Punteggio** 2,00/2,00

**Valutazione** 10,00 su un massimo di 10,00 (100%)

**Domanda 1**

Risposta corretta

Punteggio ottenuto 1,00 su 1,00

Quando aggiungo in testa ad una lista semplice, l'indirizzo del primo elemento della lista.....

Scegli un'alternativa:

- a. **non cambia**: il passaggio per riferimento del parametro corrispondente all'indirizzo di inizio lista viene fatto solo per ragioni di efficienza
- b. **cambia**: il nuovo indirizzo sarà l'indirizzo della cella creata all'interno della funzione "insert". E' per questo che dobbiamo ✓ Giusto passare il parametro formale corrispondente all'indirizzo di inizio lista, per riferimento
- c. **a volte cambia, a volte no**: la funzione "insert" è infatti una funzione randomizzata

Risposta corretta.

La risposta corretta è: **cambia**: il nuovo indirizzo sarà l'indirizzo della cella creata all'interno della funzione "insert". E' per questo che dobbiamo passare il parametro formale corrispondente all'indirizzo di inizio lista, per riferimento

**Domanda 2**

Risposta corretta

Punteggio ottenuto 1,00 su 1,00

Quando scandisco una lista semplice "s", dal primo elemento all'ultimo, la condizione per verificare di avere raggiunto l'ultimo elemento della lista è.....

Scegli un'alternativa:

- a. che il puntatore a next dell'elemento corrente sia nullptr (o molto meglio emptyList, se ho definito una costante apposita) ✓ Vero
- b. che il puntatore a next dell'elemento corrente sia "s" (ovvero il puntatore all'inizio della lista)
- c. che il payload (o "info", o "elem", a seconda di come abbiamo chiamato i campi della struct) sia uguale a un valore convenzionale, ad esempio -100000

Risposta corretta.

La risposta corretta è: che il puntatore a next dell'elemento corrente sia nullptr (o molto meglio emptyList, se ho definito una costante apposita)

**Iniziato** lunedì, 6 marzo 2023, 20:37

**Stato** Completato

**Terminato** lunedì, 6 marzo 2023, 20:37

**Tempo impiegato** 23 secondi

**Punteggio** 2,00/2,00

**Valutazione** 10,00 su un massimo di 10,00 (100%)

**Domanda 1**

Risposta corretta

Punteggio ottenuto 1,00 su 1,00

Quando aggiungo un elemento in testa ad una lista circolare, l'indirizzo del primo elemento della lista.....

Scegli un'alternativa:

- a. **non cambia**: il passaggio per riferimento del parametro corrispondente all'indirizzo di inizio lista viene fatto solo per ragioni di efficienza
- b. **cambia**: il nuovo indirizzo sarà l'indirizzo della cella creata all'interno della funzione "insert". E' per questo che dobbiamo ✓ Giusto passare il parametro formale corrispondente all'indirizzo di inizio lista, per riferimento. Inoltre, devo anche preoccuparmi di aggiornare il "next" dell'ultimo elemento, che adesso punterà a quello appena inserito
- c. **a volte cambia, a volte no**: la funzione "insert" è infatti una funzione randomizzata

Risposta corretta.

La risposta corretta è: **cambia**: il nuovo indirizzo sarà l'indirizzo della cella creata all'interno della funzione "insert". E' per questo che dobbiamo passare il parametro formale corrispondente all'indirizzo di inizio lista, per riferimento. Inoltre, devo anche preoccuparmi di aggiornare il "next" dell'ultimo elemento, che adesso punterà a quello appena inserito

**Domanda 2**

Risposta corretta

Punteggio ottenuto 1,00 su 1,00

Quando scandisco una lista circolare "s", dal primo elemento all'ultimo, la condizione per verificare di avere raggiunto l'ultimo elemento della lista è.....

Scegli un'alternativa:

- a. che il payload (o "info", o "elem", a seconda di come abbiamo chiamato i campi della struct) sia uguale a un valore convenzionale, ad esempio -100000
- b. che il puntatore a next dell'elemento corrente sia "s" (ovvero il puntatore all'inizio della lista) ✓ Giusto
- c. che il puntatore a next dell'elemento corrente sia nullptr (o molto meglio emptyList, se ho definito una costante apposita)

Risposta corretta.

La risposta corretta è: che il puntatore a next dell'elemento corrente sia "s" (ovvero il puntatore all'inizio della lista)

**Iniziato** lunedì, 6 marzo 2023, 20:37

**Stato** Completato

**Terminato** lunedì, 6 marzo 2023, 20:38

**Tempo impiegato** 27 secondi

**Punteggio** 1,00/1,00

**Valutazione** 10,00 su un massimo di 10,00 (100%)

**Domanda 1**

Risposta corretta

Punteggio ottenuto 1,00 su 1,00

**La vostra risposta non verrà valutata e non concorre in alcun modo al voto!**

Si considerino le seguenti affermazioni: una sola è corretta. Indicare quale.

Scegli un'alternativa:

- a. ad ASD abbiamo supportato encapsulation e information hiding mediante l'uso di namespace; esistono approcci migliori e più eleganti, ad esempio la programmazione ad oggetti che studierete a "LPO" il prossimo anno Vero
- b. ad ASD abbiamo supportato encapsulation e information hiding mediante l'uso di namespace; questo è quanto di meglio la comunità informatica abbia inventato fino a questo momento e non esistono approcci migliori
- c. i tipi di dato non hanno niente a che fare con encapsulation e information hiding
- d. ad ASD non ci siamo mai posti il problema di cosa siano e di come supportare encapsulation e information hiding
- e. l'information hiding è una pratica obsoleta: le moderne tecniche di ingegneria del software prevedono di non nascondere più il funzionamento interno – deciso in fase di progetto – di una parte di un programma, ma di rendere qualunque dato, funzione e dettaglio implementativo accessibile a qualunque applicazione ed utente, per ragioni di sicurezza e modularità

Risposta corretta.

La risposta corretta è:

ad ASD abbiamo supportato encapsulation e information hiding mediante l'uso di namespace; esistono approcci migliori e più eleganti, ad esempio la programmazione ad oggetti che studierete a "LPO" il prossimo anno

<b>Iniziato</b>	lunedì, 6 marzo 2023, 20:38
<b>Stato</b>	Completato
<b>Terminato</b>	lunedì, 6 marzo 2023, 20:38
<b>Tempo impiegato</b>	10 secondi
<b>Punteggio</b>	1,00/1,00
<b>Valutazione</b>	<b>10,00</b> su un massimo di 10,00 ( <b>100%</b> )

**Domanda 1**

Risposta corretta

Punteggio ottenuto 1,00 su 1,00

Qual è il significato dell'operatore % in C++

Scegli un'alternativa:

- a. Calcola il "modulo", ovvero il resto della divisione intera ✓ Vero
- b. Calcola la percentuale
- c. Calcola lo shift bit a bit a destra

Risposta corretta.

La risposta corretta è: Calcola il "modulo", ovvero il resto della divisione intera

**Iniziato** lunedì, 20 marzo 2023, 09:29

**Stato** Completato

**Terminato** lunedì, 20 marzo 2023, 09:29

**Tempo impiegato** 31 secondi

**Punteggio** 1,00/1,00

**Valutazione** 10,00 su un massimo di 10,00 (100%)

### Domanda 1

Risposta corretta

Punteggio ottenuto 1,00 su 1,00

Supponendo che il file header includa la direttiva

`const List EMPTY_SEQ = NULL;`

dove List è implementata come descritto nelle slide, qual è il modo corretto di implementare la funzione

`bool isEmpty(const List& l)`

che restituisce true se la lista è vuota e false altrimenti?

Scegli un'alternativa:

a. `bool isEmpty(const List& l)`

```
{
    if (l == EMPTY_SEQ) return true; else return false;
}
```

b. `bool isEmpty(const List& l)`

```
{
    return (l == EMPTY_SEQ);
}
```

c. `bool isEmpty(const List& l)`

```
{
    return (l == NULL);
}
```

Risposta corretta.

La risposta corretta è:

```
bool isEmpty(const List& l)
{
    return (l == EMPTY_SEQ);
}
```

**Iniziato** lunedì, 20 marzo 2023, 09:30

**Stato** Completato

**Terminato** lunedì, 20 marzo 2023, 09:32

**Tempo impiegato** 2 min. 4 secondi

**Punteggio** 1,00/1,00

**Valutazione** 10,00 su un massimo di 10,00 (100%)

**Domanda 1**

Risposta corretta

Punteggio ottenuto 1,00 su 1,00

Quale delle seguenti affermazioni è vera, rispetto alla funzione size descritta nelle slide, nel codice e nei podcast?

Scegli un'alternativa:

- a. La funzione size in alcuni casi può accedere alla componente "next" di una struct puntata da NULL: questo, quando accade, non crea alcun problema
- b. La funzione size in alcuni casi può accedere alla componente "next" di una struct puntata da NULL: questo, quando accade, è motivo di segmentation fault
- c. La funzione size è implementata correttamente e il controllo isEmpty(l) evita che si acceda l->next, nel caso in cui l sia NULL ✓

Risposta corretta.

La risposta corretta è: La funzione size è implementata correttamente e il controllo isEmpty(l) evita che si acceda l->next, nel caso in cui l sia NULL

**Iniziato** lunedì, 20 marzo 2023, 09:32

**Stato** Completato

**Terminato** lunedì, 20 marzo 2023, 09:32

**Tempo impiegato** 13 secondi

**Punteggio** 1,00/1,00

**Valutazione** 10,00 su un massimo di 10,00 (100%)

**Domanda 1**

Risposta corretta

Punteggio ottenuto 1,00 su 1,00

Una variabile di tipo puntatore....

Scegli un'alternativa:

- a. ...ha una lunghezza variabile, a seconda di cosa viene puntato
- b. ...può essere passata per riferimento, se il suo valore viene modificato nel corpo della funzione e questa modifica deve essere visibile anche quando la funzione ha terminato la sua esecuzione ✓ Vero
- c. ...non può essere passata per riferimento

Risposta corretta.

La risposta corretta è: ...può essere passata per riferimento, se il suo valore viene modificato nel corpo della funzione e questa modifica deve essere visibile anche quando la funzione ha terminato la sua esecuzione

**Iniziato** lunedì, 20 marzo 2023, 09:32

**Stato** Completato

**Terminato** lunedì, 20 marzo 2023, 09:33

**Tempo impiegato** 1 min.

**Punteggio** 1,00/1,00

**Valutazione** 10,00 su un massimo di 10,00 (100%)

**Domanda 1**

Risposta corretta

Punteggio ottenuto 1,00 su 1,00

Quale delle seguenti affermazioni è corretta?

Scegli un'alternativa:

- a. Non me ne importa niente delle Leggi di De Morgan, questo è ASD, mica un corso di logica
- b. sia P una proprietà degli elementi dell'insieme S; per ogni x appartenente a S vale P(x) EQUIVALE LOGICAMENTE A not esiste x appartenente a S per cui NOT vale P(x) ✓ Vero
- c. sia P una proprietà degli elementi dell'insieme S; per ogni x appartenente a S vale P(x) EQUIVALE LOGICAMENTE A not esiste x appartenente a S per cui vale P(x)

Risposta corretta.

La risposta corretta è: sia P una proprietà degli elementi dell'insieme S; per ogni x appartenente a S vale P(x) EQUIVALE LOGICAMENTE A not esiste x appartenente a S per cui NOT vale P(x)

<b>Iniziato</b>	sabato, 11 marzo 2023, 10:46
<b>Stato</b>	Completato
<b>Terminato</b>	sabato, 11 marzo 2023, 10:47
<b>Tempo impiegato</b>	40 secondi
<b>Punteggio</b>	1,00/1,00
<b>Valutazione</b>	<b>10,00</b> su un massimo di 10,00 ( <b>100%</b> )

**Domanda 1**

Risposta corretta

Punteggio ottenuto 1,00 su 1,00

**Rispetto alla domanda presentata nelle slide:**

*Devi implementare un programma che necessita di tre funzioni f1, f2, f3, oltre alla funzione di lettura dell'input da file e al main.*

*f3 richiama f1 ed f2.*

*Tutte le funzioni operano su liste di interi, letti da file, e devono dare risultati diversi a seconda che la lista passata come argomento*

- 1.sia vuota
- 2.non sia vuota
- 3.contenga solo numeri pari
- 4.contenga solo numeri dispari
- 5.contenga più numeri pari che numeri dispari
- 6.tutti gli altri casi

**Cosa fate?**

Scegli un'alternativa:

- a. Scrivete tutto il codice guardandovi bene dal compilare di tanto in tanto.
- b. Ragionate, progettate, implementate e compilate, ma il primo run lo fate solo quando avete finito la stesura (e verifica della compilazione) di tutto il programma
- c. Svolgete i seguenti passi:

Bravi!

1. Ragionate, progettate, implementate e compilate per prima la funzione che legge da file, creando una primissima versione in cui stampate su standard output ogni elemento letto dal file, invece che inserirlo nella lista, per essere sicuri che la scansione degli elementi del file sia corretta.
2. Ragionate, progettate, implementate e compilate la funzione di inserimento di un intero nella lista e verificate il suo funzionamento facendone diverse chiamate dal main, con interi diversi, e verificando mediante una stampa della lista che la funzione operi correttamente.
3. Raffinate la funzione di lettura da file in modo che gli elementi letti vengano inseriti nella lista, usando la funzione "addElem" implementata al passo 2. Testate la funzione facendo una stampa del contenuto della lista letto.
4. Ragionate, progettate, implementate e compilate f1 e la testate su almeno 6 input diversi, corrispondenti ai 6 casi diversi che f1 deve prendere in considerazione. Ancora meglio se la testate su 12 o 18 input....
5. Stessa cosa con f2.
6. Infine Ragionate, progettate, implementate e compilate f3, che fa uso di f1 ed f2 (ma a questo punto, siete ragionevolmente certo che f1 ed f2 funzionino, quindi se riscontrate degli errori saranno in f3), e la testate su almeno 6 input diversi.
7. Implementate il main definitivo rimuovendo le chiamate di debug alle varie funzioni che avete implementato e lo testate adeguatamente su un numero ragionevole di casi.

**Risposta corretta.**

La risposta corretta è: Svolgete i seguenti passi:

1. Ragionate, progettate, implementate e compilate per prima la funzione che legge da file, creando una primissima versione in cui stampate su standard output ogni elemento letto dal file, invece che inserirlo nella lista, per essere sicuri che la scansione degli elementi del file sia corretta.
2. Ragionate, progettate, implementate e compilate la funzione di inserimento di un intero nella lista e verificate il suo funzionamento facendone diverse chiamate dal main, con interi diversi, e verificando mediante una stampa della lista che la funzione operi correttamente.
3. Raffinate la funzione di lettura da file in modo che gli elementi letti vengano inseriti nella lista, usando la funzione "addElem" implementata al

passo 2. Testate la funzione facendo una stampa del contenuto della lista letto.

4. Ragionate, progettate, implementate e compilate f1 e la testate su almeno 6 input diversi, corrispondenti ai 6 casi diversi che f1 deve prendere in considerazione. Ancora meglio se la testate su 12 o 18 input....

5. Stessa cosa con f2.

6. Infine Ragionate, progettate, implementate e compilate f3, che fa uso di f1 ed f2 (ma a questo punto, siete ragionevolmente certo che f1 ed f2 funzionino, quindi se riscontrate degli errori saranno in f3), e la testate su almeno 6 input diversi.

7. Implementate il main definitivo rimuovendo le chiamate di debug alle varie funzioni che avete implementato e lo testate adeguatamente su un numero ragionevole di casi.

**Iniziato** lunedì, 20 marzo 2023, 09:35

**Stato** Completato

**Terminato** lunedì, 20 marzo 2023, 09:35

**Tempo impiegato** 10 secondi

**Punteggio** 2,00/2,00

**Valutazione** 10,00 su un massimo di 10,00 (100%)

**Domanda 1**

Risposta corretta

Punteggio ottenuto 1,00 su 1,00

Nella funzione merge...

Scegli un'alternativa:

- a. Non serve alcuna struttura dati ausiliaria: merge è una funzione che opera "in place"
- b. Ho bisogno di due strutture dati ausiliarie per copiare le due parti della porzione di vettore (o array, o sequenza) che sono state riordinate grazie alle precedenti chiamate di ms, e che ora vanno fuse ("merge")

 Giusto

Risposta corretta.

La risposta corretta è: Ho bisogno di due strutture dati ausiliarie per copiare le due parti della porzione di vettore (o array, o sequenza) che sono state riordinate grazie alle precedenti chiamate di ms, e che ora vanno fuse ("merge")

**Domanda 2**

Risposta corretta

Punteggio ottenuto 1,00 su 1,00

Se all'esame si chiede lo pseudo-codice di merge...

Scegli un'alternativa:

- a. ...è fondamentale scrivere anche lo pseudo-codice per i casi in cui la scansione di una delle due sotto-sequenze sia terminata, e gli elementi dell'altra vadano copiati nella sequenza risultato. La mancanza di questa parte dello pseudo-codice invalida la risposta.
- b. ...è fondamentale scrivere il "cuore" dell'algoritmo, in cui gli elementi delle sottosequenze vengono confrontati due a due ✓ Giusto ed il "vincitore" (il minore) viene copiato nella sequenza risultante. Si può omettere il codice relativo a cosa succede quando una delle due sottosequenze è terminata, ma bisogna almeno scrivere "si trascurano i casi estremi in cui una delle due sottosequenze è stata scandita completamente"

Risposta corretta.

La risposta corretta è: ...è fondamentale scrivere il "cuore" dell'algoritmo, in cui gli elementi delle sottosequenze vengono confrontati due a due ed il "vincitore" (il minore) viene copiato nella sequenza risultante. Si può omettere il codice relativo a cosa succede quando una delle due sottosequenze è terminata, ma bisogna almeno scrivere "si trascurano i casi estremi in cui una delle due sottosequenze è stata scandita completamente"

<b>Iniziato</b>	lunedì, 20 marzo 2023, 09:36
<b>Stato</b>	Completato
<b>Terminato</b>	lunedì, 20 marzo 2023, 09:36
<b>Tempo impiegato</b>	27 secondi
<b>Punteggio</b>	1,00/1,00
<b>Valutazione</b>	10,00 su un massimo di 10,00 (100%)

**Domanda 1**

Risposta corretta

Punteggio ottenuto 1,00 su 1,00

ES4: Supponiamo che l'assegnazione “=” costi c1, il confronto “<” e “==” costino c2, l'incremento “++” costi c3, la stampa “cout<<” costi c4 e la lettura da stdin “cin>>” costi c5.

```
int n;
cin >> n;
for (int i=0; i<n; ++i)
    for (int j=0; j<n; ++j)
        cout << "(" << i << "," << j << ")\n";
```

Scegli un'alternativa:

- a. Sarà proporzionale a n, come per gli altri esempi visti prima dove c'era un ciclo for con i che andava da 0 a n. Anche qui abbiamo un for esterno simile, quindi il numero di operazioni sarà simile. Ma a calcolare il numero esatto di c1, c2, c3, ...., non ce la faccio proprio....
- b. Sarà proporzionale a log<sub>2</sub> n, ho visto che da qualche parte nelle slide i logaritmi sono comparsi prima o poi serviranno. Ma a calcolare il numero esatto di c1, c2, c3, ...., non ce la faccio proprio....
- c. Sarà proporzionale a n<sup>2</sup>, visto che per ogni i (che va da 0 a n, quindi per circa n oppure n+1 volte) ripeto circa n operazioni (dovute al for annidato internamente, dove ho “per ogni j, con j che va da 0 a n”). Quindi ho circa n \* n operazioni (quelle del ciclo esterno, per quelle del ciclo interno). Ma a calcolare il numero esatto di c1, c2, c3, ...., non ce la faccio proprio....

✓ Giusto! Nessuno ce la fa a calcolare i c1, c2, c3... e nessuno lo farà...

Risposta corretta.

La risposta corretta è: Sarà proporzionale a n<sup>2</sup>, visto che per ogni i (che va da 0 a n, quindi per circa n oppure n+1 volte) ripeto circa n operazioni (dovute al for annidato internamente, dove ho “per ogni j, con j che va da 0 a n”). Quindi ho circa n \* n operazioni (quelle del ciclo esterno, per quelle del ciclo interno). Ma a calcolare il numero esatto di c1, c2, c3, ...., non ce la faccio proprio....

**Iniziato** lunedì, 20 marzo 2023, 09:37

**Stato** Completato

**Terminato** lunedì, 20 marzo 2023, 09:37

**Tempo impiegato** 43 secondi

**Punteggio** 1,00/1,00

**Valutazione** 10,00 su un massimo di 10,00 (100%)

### Domanda 1

Risposta corretta

Punteggio ottenuto 1,00 su 1,00

Perché è possibile togliere la base del logaritmo e semplificare, ad esempio:

$7n \log_2 n + 7n = n \log n$

(ovvero, stiamo affermando che  $7n \log_2 n + 7n$  appartiene a  $\theta(n \log n)$ ) ?

Scegli un'alternativa:

- a. La risposta non è mai stata data: i docenti danno per scontato che abbiamo tutti acquisito le nozioni di matematica di base e, giustamente, non fanno alcun ripasso
- b. La risposta è stata data all'inizio della seconda settimana: i logaritmi sono di ordine inferiore per definizione, quindi posso trascurare non solo la base del logaritmo, ma il logaritmo stesso
- c. La risposta è stata data il primo giorno di lezione, nel ripasso dei logaritmi: si può sempre cambiare la base del logaritmo ✓ Giusto moltiplicando per una costante. Visto che quando calcoliamo la complessità degli algoritmi le costanti moltiplicative non ci interessano, anche la base del logaritmo diventa irrilevante e la possiamo trascurare

Risposta corretta.

La risposta corretta è: La risposta è stata data il primo giorno di lezione, nel ripasso dei logaritmi: si può sempre cambiare la base del logaritmo moltiplicando per una costante. Visto che quando calcoliamo la complessità degli algoritmi le costanti moltiplicative non ci interessano, anche la base del logaritmo diventa irrilevante e la possiamo trascurare

**Iniziato** lunedì, 20 marzo 2023, 09:37

**Stato** Completato

**Terminato** lunedì, 20 marzo 2023, 09:39

**Tempo impiegato** 1 min. 53 secondi

**Punteggio** 1,00/1,00

**Valutazione** 10,00 su un massimo di 10,00 (100%)

**Domanda 1**

Risposta corretta

Punteggio ottenuto 1,00 su 1,00

La complessità del frammento di codice mostrato nell'esercizio ES5, considerando i principi del calcolo di complessità (trascurare costanti moltiplicative, trascurare termini di ordine inferiore) è....

Scegli un'alternativa:

- a.  $\theta(n^2)$  ✓ giusto
- b.  $\theta(n)$
- c.  $\theta(1/2n^2 + 1/2n)$

Risposta corretta.

La risposta corretta è:  $\theta(n^2)$

**Iniziato** lunedì, 20 marzo 2023, 09:39

**Stato** Completato

**Terminato** lunedì, 20 marzo 2023, 09:41

**Tempo impiegato** 1 min. 58 secondi

**Punteggio** 5,00/5,00

**Valutazione** 10,00 su un massimo di 10,00 (100%)

#### Domanda 1

Risposta corretta

Punteggio ottenuto 1,00 su 1,00

La complessità del frammento di codice ES6 è....

Scegli un'alternativa:

- a. theta( n^2 ) sia nel caso migliore che in quello peggiore
- b. theta( n ) sia nel caso migliore che in quello peggiore
- c. theta( 1 ), ovvero complessità costante sia nel caso migliore che in quello peggiore ✓

Risposta corretta.

La risposta corretta è: theta( 1 ), ovvero complessità costante sia nel caso migliore che in quello peggiore

#### Domanda 2

Risposta corretta

Punteggio ottenuto 1,00 su 1,00

Qual è la complessità della funzione print\_list chiamata su una lista list che contiene n elementi?

Scegli un'alternativa:

- a. theta ( 1 ) sia nel caso migliore che nel caso peggiore
- b. theta ( n ) sia nel caso migliore che nel caso peggiore ✓
- c. theta ( n^2 ) sia nel caso migliore che nel caso peggiore

Risposta corretta.

La risposta corretta è: theta ( n ) sia nel caso migliore che nel caso peggiore

**Domanda 3**

Risposta corretta

Punteggio ottenuto 1,00 su 1,00

Sia  $n$  la dimensione della lista list; qual è la complessità di ES7?

Scegli un'alternativa:

- a.  $\theta(n^2)$  sia nel caso migliore che in quello peggiore
- b.  $\theta(1)$  sia nel caso migliore che in quello peggiore ✓
- c.  $\theta(n)$  sia nel caso migliore che in quello peggiore

Risposta corretta.

La risposta corretta è:  $\theta(1)$  sia nel caso migliore che in quello peggiore

**Domanda 4**

Risposta corretta

Punteggio ottenuto 1,00 su 1,00

Qual è la complessità di read\_list chiamata su uno stream di  $n$  elementi, dove l'inserimento dell'elemento nella lista avviene in testa?

Scegli un'alternativa:

- a.  $\theta(1)$  sia nel caso migliore che nel caso peggiore
- b.  $\theta(n^2)$  sia nel caso migliore che nel caso peggiore
- c.  $\theta(n)$  sia nel caso migliore che nel caso peggiore ✓

Risposta corretta.

La risposta corretta è:  $\theta(n)$  sia nel caso migliore che nel caso peggiore

**Domanda 5**

Risposta corretta

Punteggio ottenuto 1,00 su 1,00

Nell'esercizio 3, la docente ha insistito sul fatto che la complessità degli algoritmi va calcolata nel caso peggiore e nel caso migliore (a volte anche nel caso medio), perché spesso la complessità è diversa in questi due casi: perché non ha più specificato se stavamo ragionando nel caso migliore/peggiore negli altri esercizi?

Scegli un'alternativa:

- a. Perché in tutti gli algoritmi visti, il caso migliore è theta( $n^2$ ): trattandosi di un default, non serve esplicitarlo
- b. Perché non è vero che la complessità va calcolata in un caso migliore/peggiore/medio
- c. Perché si è dimenticata di dirlo.... ma la ragione per cui si è dimenticata è che negli esercizi visti fino ad ora, a parte il #3, il caso migliore e peggiore coincidono o, detto in altro modo, non esistono un caso migliore ed uno peggiore: l'algoritmo ha la stessa complessità in ogni caso, ovviamente espressa in funzione di n



Risposta corretta.

La risposta corretta è: Perché si è dimenticata di dirlo.... ma la ragione per cui si è dimenticata è che negli esercizi visti fino ad ora, a parte il #3, il caso migliore e peggiore coincidono o, detto in altro modo, non esistono un caso migliore ed uno peggiore: l'algoritmo ha la stessa complessità in ogni caso, ovviamente espressa in funzione di n

**Iniziato** lunedì, 20 marzo 2023, 09:42

**Stato** Completato

**Terminato** lunedì, 20 marzo 2023, 09:42

**Tempo impiegato** 6 secondi

**Punteggio** 1,00/1,00

**Valutazione** 10,00 su un massimo di 10,00 (100%)

### Domanda 1

Risposta corretta

Punteggio ottenuto 1,00 su 1,00

Qual è la complessità di `read_list` su uno stream di  $n$  elementi, supponendo che `last_insert` inserisca un elemento in coda alla lista `list` e che `list` sia una lista collegata semplicemente e con sentinella (**il fatto che `list` sia circolare o no, non cambia la complessità: cambierebbe solo se fosse circolare e doppiamente collegata**)

Scegli un'alternativa:

- a. Theta ( 1 ) nel caso migliore e Theta (  $n$  ) nel caso peggiore
- b. Theta (  $n$  ) nel caso migliore e peggiore
- c. Theta (  $\checkmark$   $n^2$  ) nel caso migliore e peggiore: Esatto, è Theta (  $n^2$  ) nel caso migliore e peggiore: la presenza della sommatoria ci riconduce al ragionamento fatto per ESS; inoltre, non vale dire che se lo stream è vuoto, la complessità è costante. Questa informazione è già catturata dal fatto che se  $n == 0$ , la complessità è Theta (  $0^2$  ) = Theta ( 0 ) che sta ad indicare, come Theta ( 1 ), una complessità costante. Non serve discriminare due casi migliore/peggiore, se sono già entrambi catturati da un'unica espressione.

Risposta corretta.

La risposta corretta è: Theta (  $n^2$  ) nel caso migliore e peggiore

<b>Iniziato</b>	lunedì, 20 marzo 2023, 09:42
<b>Stato</b>	Completato
<b>Terminato</b>	lunedì, 20 marzo 2023, 09:43
<b>Tempo impiegato</b>	32 secondi
<b>Punteggio</b>	1,00/1,00
<b>Valutazione</b>	10,00 su un massimo di 10,00 (100%)

**Domanda 1**

Risposta corretta

Punteggio ottenuto 1,00 su 1,00

Qual è la complessità di insertElem così come implementata in ES16, ovvero con controllo che l'elemento non sia già presente, nel qual caso non va re-inserito

Scegli un'alternativa:

- a. La complessità di insertElem è dominata dalla complessità di member e vale Theta ( n ) nel caso migliore, corrispondente al caso migliore di member (l'elemento da inserire c'è già e viene trovato subito all'inizio della lista) e Theta ( 1 ) nel caso peggiore, corrispondente al caso peggiore di member in cui l'elemento non c'è, ma per scoprire che non c'è va scandita l'intera sequenza
- b. Theta ( 1 ) nel caso migliore e peggiore: abbiamo già visto che l'inserimento in testa in una lista semplice ha complessità costante
- c. La complessità di insertElem è dominata dalla complessità di member e vale Theta ( 1 ) nel caso migliore, corrispondente al caso migliore di member (l'elemento da inserire c'è già e viene trovato subito all'inizio della lista) e Theta ( n ) nel caso peggiore, corrispondente al caso peggiore di member in cui l'elemento non c'è, ma per scoprire che non c'è va scandita l'intera sequenza

Risposta corretta.

La risposta corretta è: La complessità di insertElem è dominata dalla complessità di member e vale Theta ( 1 ) nel caso migliore, corrispondente al caso migliore di member (l'elemento da inserire c'è già e viene trovato subito all'inizio della lista) e Theta ( n ) nel caso peggiore, corrispondente al caso peggiore di member in cui l'elemento non c'è, ma per scoprire che non c'è va scandita l'intera sequenza

**Iniziato** lunedì, 20 marzo 2023, 09:43

**Stato** Completato

**Terminato** lunedì, 20 marzo 2023, 09:43

**Tempo impiegato** 12 secondi

**Punteggio** 1,00/1,00

**Valutazione** 10,00 su un massimo di 10,00 (100%)

**Domanda 1**

Risposta corretta

Punteggio ottenuto 1,00 su 1,00

Con Big-Oh( $g(n)$ ) si denota...

Scegli un'alternativa:

- a. La complessità di un algoritmo nel caso peggiore
- b. La funzione  $g(n)$
- c. La classe delle funzioni che "da un certo  $n_0 \geq 0$  in poi" sono dominate da  $c * g(n)$ , con  $c$  costante moltiplicativa reale  $> 0$  (questa  $c$  non è uguale per ogni funzione nella classe, ogni funzione potrà richiedere una costante diversa per essere dominata da  $c * g(n)$  "da un certo punto in poi") ✓

Risposta corretta.

La risposta corretta è: La classe delle funzioni che "da un certo  $n_0 \geq 0$  in poi" sono dominate da  $c * g(n)$ , con  $c$  costante moltiplicativa reale  $> 0$  (questa  $c$  non è uguale per ogni funzione nella classe, ogni funzione potrà richiedere una costante diversa per essere dominata da  $c * g(n)$  "da un certo punto in poi")

**Iniziato** lunedì, 20 marzo 2023, 09:43

**Stato** Completato

**Terminato** lunedì, 20 marzo 2023, 09:44

**Tempo impiegato** 8 secondi

**Punteggio** 1,00/1,00

**Valutazione** 10,00 su un massimo di 10,00 (100%)

**Domanda 1**

Risposta corretta

Punteggio ottenuto 1,00 su 1,00

Con Omega( $g(n)$ ) si denota...

Scegli un'alternativa:

- a. La complessità di un algoritmo nel caso migliore
- b. La funzione  $g(n)$
- c. La classe delle funzioni che "da un certo  $n_0 \geq 0$  in poi" dominano  $c * g(n)$ , con  $c$  costante moltiplicativa reale  $> 0$  (questa  $c$  non è uguale per ogni funzione nella classe, ogni funzione potrà richiedere una costante diversa per dominare  $c * g(n)$  "da un certo punto in poi")

Risposta corretta.

La risposta corretta è: La classe delle funzioni che "da un certo  $n_0 \geq 0$  in poi" dominano  $c * g(n)$ , con  $c$  costante moltiplicativa reale  $> 0$  (questa  $c$  non è uguale per ogni funzione nella classe, ogni funzione potrà richiedere una costante diversa per dominare  $c * g(n)$  "da un certo punto in poi")

**Iniziato** lunedì, 20 marzo 2023, 09:44

**Stato** Completato

**Terminato** lunedì, 20 marzo 2023, 09:44

**Tempo impiegato** 8 secondi

**Punteggio** 1,00/1,00

**Valutazione** 10,00 su un massimo di 10,00 (100%)

**Domanda 1**

Risposta corretta

Punteggio ottenuto 1,00 su 1,00

Con  $\Theta(g(n))$  si denota...

Scegli un'alternativa:

- a. La funzione  $g(n)$
- b. La classe delle funzioni che appartengono all'intersezione tra  $\Theta(g(n))$  e  $\Omega(g(n))$  ✓
- c. La complessità di un algoritmo nel caso medio

Risposta corretta.

La risposta corretta è: La classe delle funzioni che appartengono all'intersezione tra  $\Theta(g(n))$  e  $\Omega(g(n))$

**Iniziato** lunedì, 20 marzo 2023, 09:44

**Stato** Completato

**Terminato** lunedì, 20 marzo 2023, 09:45

**Tempo impiegato** 47 secondi

**Punteggio** 3,00/3,00

**Valutazione** 10,00 su un massimo di 10,00 (100%)

### Domanda 1

Risposta corretta

Punteggio ottenuto 1,00 su 1,00

Sia  $n$  la lunghezza di A. Qual è la complessità dell'algoritmo descritto in E23?

Scegli un'alternativa:

- a. Theta ( 1 ) nel caso migliore in cui i primi due elementi siano identici; Theta (  $n$  ) nel caso in cui non ci siano elementi duplicati
- b. Theta (  $n^2$  ) nel caso migliore e peggiore
- c. Theta ( 1 ) nel caso migliore in cui i primi due elementi siano identici; Theta (  $n^2$  ) nel caso in cui non ci siano elementi duplicati ✓

Risposta corretta.

La risposta corretta è: Theta ( 1 ) nel caso migliore in cui i primi due elementi siano identici; Theta (  $n^2$  ) nel caso in cui non ci siano elementi duplicati

### Domanda 2

Risposta corretta

Punteggio ottenuto 1,00 su 1,00

Sia  $n$  la lunghezza di A ed  $m$  la lunghezza di B. Qual è la complessità dell'algoritmo descritto nell'ES 22?

Scegli un'alternativa:

- a. Theta (  $n * m$  ) ✓ **Giusto:** dati due array A e B non ordinati, verificare se hanno un elemento in comune richiede di confrontare ogni elemento di A con ogni elemento di B finché non ne trovo due uguali; se il primo elemento di B è uguale al primo elemento di A sono nel caso migliore e non devo fare ulteriori controlli. La complessità è Theta( 1 ) nel caso migliore, Theta(  $n * m$  ) nel caso peggiore in cui non ci sono elementi in comune e tutti i confronti vanno fatti.
- b. Theta (  $n^2 * m^2$  ) nel caso migliore e peggiore
- c. Theta (  $n + m$  ) nel caso migliore e peggiore

Risposta corretta.

La risposta corretta è: Theta (  $n * m$  ) nel caso peggiore e Theta( 1 ) nel caso migliore

**Domanda 3**

Risposta corretta

Punteggio ottenuto 1,00 su 1,00

Se so che l'array è ordinato, qual è l'algoritmo migliore per cercare un elemento?

Scegli un'alternativa:

- a. Ricerca binaria ✓
- b. Scandire dal primo elemento fino all'ultimo
- c. Scandire dall'ultimo elemento fino al primo

Risposta corretta.

La risposta corretta è: Ricerca binaria

**Iniziato** lunedì, 20 marzo 2023, 09:45

**Stato** Completato

**Terminato** lunedì, 20 marzo 2023, 09:45

**Tempo impiegato** 22 secondi

**Punteggio** 1,00/1,00

**Valutazione** 10,00 su un massimo di 10,00 (100%)

**Domanda 1**

Risposta corretta

Punteggio ottenuto 1,00 su 1,00

La complessità di selection sort è...

Scegli un'alternativa:

- a. Theta( n^2 ) nel caso migliore e peggiore ✓
- b. Theta ( n ) nel caso migliore e Theta(n^2) nel caso peggiore
- c. Theta ( n ) nel caso migliore e peggiore

Risposta corretta.

La risposta corretta è: Theta( n^2 ) nel caso migliore e peggiore

**Iniziato** lunedì, 20 marzo 2023, 09:45

**Stato** Completato

**Terminato** lunedì, 20 marzo 2023, 09:45

**Tempo impiegato** 7 secondi

**Punteggio** 1,00/1,00

**Valutazione** 10,00 su un massimo di 10,00 (100%)

**Domanda 1**

Risposta corretta

Punteggio ottenuto 1,00 su 1,00

La complessità di insertion sort è...

Scegli un'alternativa:

- a. Theta( n^2 ) nel caso migliore e peggiore
- b. Theta ( n ) nel caso migliore e peggiore
- c. Theta ( n ) nel caso migliore e Theta(n^2) nel caso peggiore ✓

Risposta corretta.

La risposta corretta è: Theta ( n ) nel caso migliore e Theta(n^2) nel caso peggiore

**Iniziato** lunedì, 20 marzo 2023, 09:45

**Stato** Completato

**Terminato** lunedì, 20 marzo 2023, 09:45

**Tempo impiegato** 9 secondi

**Punteggio** 1,00/1,00

**Valutazione** 10,00 su un massimo di 10,00 (100%)

**Domanda 1**

Risposta corretta

Punteggio ottenuto 1,00 su 1,00

La complessità di bubble sort è...

Scegli un'alternativa:

- a. Theta ( n ) nel caso migliore e peggiore
- b. Theta( n<sup>2</sup> ) nel caso migliore e peggiore
- c. Theta ( n ) nel caso migliore e Theta(n<sup>2</sup>) nel caso peggiore ✓

Risposta corretta.

La risposta corretta è: Theta ( n ) nel caso migliore e Theta(n<sup>2</sup>) nel caso peggiore

<b>Iniziato</b>	martedì, 18 aprile 2023, 16:00
<b>Stato</b>	Completato
<b>Terminato</b>	martedì, 18 aprile 2023, 16:01
<b>Tempo impiegato</b>	39 secondi
<b>Punteggio</b>	1,00/1,00
<b>Valutazione</b>	<b>10,00</b> su un massimo di 10,00 ( <b>100%</b> )

**Domanda 1**

Risposta corretta

Punteggio ottenuto 1,00 su 1,00

Quale delle seguenti affermazioni sulla struttura dati "array dinamico con size, maxsize e ridimensionamento" implementata dai docenti e resa disponibile durante il corso è corretta (possono esserci più risposte corrette)

Scegli una o più alternative:

- a. Il ridimensionamento dell'array dinamico può avvenire solo quando si chiama la funzione add
- b. size indica la dimensione totale attualmente allocata per l'array, maxsize indica il numero di elementi della lista (maxsize può essere minore di size, non è detto che tutto lo spazio disponibile sia usato)
- c. I vector del C++ sono implementati in modo molto simile agli array dinamici con size, maxsize e ridimensionamento ✓
- d. Il ridimensionamento dell'array dinamico può avvenire quando si chiama la funzione add e quando si chiama la funzione removePos (in questo caso può avere un ridimensionamento per diminuire la dimensione dell'array, non per aumentarla) ✓
- e. I docenti suggeriscono di usare come dimensione di BLOCKDIM almeno 40000, meglio ancora 80000, così si evita di dover ridimensionare spesso; 40000 è la dimensione ideale anche quando le liste contengono non più di un centinaio di elementi.
- f. BLOCKDIM indica la dimensione totale attualmente allocata per l'array, size indica il numero di elementi della lista (size può essere minore di BLOCKDIM, non è detto che tutto lo spazio disponibile sia usato)
- g. maxsize indica la dimensione totale attualmente allocata per l'array, size indica il numero di elementi della lista (size può essere minore di maxsize, non è detto che tutto lo spazio disponibile sia usato) ✓
- h. Nella implementazione fornita dai docenti la dimensione di BLOCKDIM è stabilita a runtime

Risposta corretta.

Le risposte corrette sono: maxsize indica la dimensione totale attualmente allocata per l'array, size indica il numero di elementi della lista (size può essere minore di maxsize, non è detto che tutto lo spazio disponibile sia usato), I vector del C++ sono implementati in modo molto simile agli array dinamici con size, maxsize e ridimensionamento, Il ridimensionamento dell'array dinamico può avvenire quando si chiama la funzione add e quando si chiama la funzione removePos (in questo caso può avere un ridimensionamento per diminuire la dimensione dell'array, non per aumentarla)



**Iniziato** lunedì, 20 marzo 2023, 09:46

**Stato** Completato

**Terminato** lunedì, 20 marzo 2023, 09:48

**Tempo impiegato** 1 min. 51 secondi

**Punteggio** 3,00/3,00

**Valutazione** 10,00 su un massimo di 10,00 (100%)

### Domanda 1

Risposta corretta

Punteggio ottenuto 1,00 su 1,00

Qual è la complessità di mergesort?

Scegli un'alternativa:

- a. Theta ( n ) nel caso migliore e peggiore
- b. Theta ( n log n ) nel caso migliore e peggiore ✓
- c. Theta ( n log n ) nel caso migliore e Theta ( n^2 ) nel caso peggiore

Risposta corretta.

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

### Domanda 2

Risposta corretta

Punteggio ottenuto 1,00 su 1,00

Nella complessità di mergesort, Theta ( n log n ), da dove viene il fattore "n" ?

Scegli un'alternativa:

- a. Dal numero di livelli dell'albero della ricorsione, che è n
- b. Dalla complessità di merge che è Theta ( n )
- c. Dal numero di operazioni svolte ad ogni livello dell'albero della ricorsione: al livello j si effettuano  $2^j$  chiamate a merge, ciascuna su una porzione di array lunga  $n/(2^j)$ . Ciascuna di queste chiamate ha complessità lineare nella dimensione della porzione su cui viene chiamata, quindi Theta (  $n/(2^j)$  ): ad ogni livello faccio  $2^j * \Theta(n/(2^j))$  operazioni, che determinano la complessità Theta( n ) ✓

Risposta corretta.

La risposta corretta è: Dal numero di operazioni svolte ad ogni livello dell'albero della ricorsione: al livello j si effettuano  $2^j$  chiamate a merge, ciascuna su una porzione di array lunga  $n/(2^j)$ . Ciascuna di queste chiamate ha complessità lineare nella dimensione della porzione su cui viene chiamata, quindi Theta (  $n/(2^j)$  ): ad ogni livello faccio  $2^j * \Theta(n/(2^j))$  operazioni, che determinano la complessità Theta( n )

**Domanda 3**

Risposta corretta

Punteggio ottenuto 1,00 su 1,00

Nella complessità di mergesort, Theta (  $n \log n$  ), da dove viene il fattore "log n" ?

Scegli un'alternativa:

- a. Dal costo della ricerca binaria effettuata per trovare la metà dell'array
- b. Dal numero dei livelli dell'albero della ricorsione ✓
- c. Dal costo di merge, che è Theta (  $\log n$  ) nel caso peggiore

Risposta corretta.

La risposta corretta è: Dal numero dei livelli dell'albero della ricorsione

<b>Iniziato</b>	martedì, 18 aprile 2023, 16:01
<b>Stato</b>	Completato
<b>Terminato</b>	martedì, 18 aprile 2023, 16:02
<b>Tempo impiegato</b>	42 secondi
<b>Punteggio</b>	1,00/1,00
<b>Valutazione</b>	<b>10,00</b> su un massimo di 10,00 ( <b>100%</b> )

**Domanda 1**

Risposta corretta

Punteggio ottenuto 1,00 su 1,00

[Domanda #1]

Sia  $f(n) = 7n^3 + 12n \log_5 n + 3 \log n + 23$ 

Selezionare tutte le risposte corrette (possono essere più di una)

Scegli una o più alternative:

- a.  $f(n) = O(n)$
- b.  $f(n) = \Theta(1)$
- c.  $f(n) = \Omega(1)$  ✓
- d.  $f(n) = O(1)$
- e.  $f(n) = \Omega(n)$  ✓
- f.  $f(n) = \Theta(n)$

Risposta corretta.

Le risposte corrette sono:  $f(n) = \Omega(1)$ ,  $f(n) = \Omega(n)$



**Iniziato** giovedì, 30 marzo 2023, 17:37

**Stato** Completato

**Terminato** giovedì, 30 marzo 2023, 17:37

**Tempo impiegato** 7 secondi

**Punteggio** 1,00/1,00

**Valutazione** 10,00 su un massimo di 10,00 (100%)

**Domanda 1**

Risposta corretta

Punteggio ottenuto 1,00 su 1,00

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 partition. Qual è la complessità di partition nel caso migliore e peggiore?

Scegli un'alternativa:

- a. Theta ( fine-inizio ) nel caso migliore e peggiore: la complessità dipende dalla dimensione della porzione di A su cui partition opera ✓
- b. Theta (  $n$  ) nel caso peggiore e Theta ( 1 ) nel caso migliore, quello in cui  $n = 1$
- c. Theta (  $n$  ) nel caso migliore e peggiore: la complessità dipende dalla dimensione di A

Risposta corretta.

La risposta corretta è: Theta ( fine-inizio ) nel caso migliore e peggiore: la complessità dipende dalla dimensione della porzione di A su cui partition opera



<b>Iniziato</b>	martedì, 18 aprile 2023, 16:02
<b>Stato</b>	Completato
<b>Terminato</b>	martedì, 18 aprile 2023, 16:03
<b>Tempo impiegato</b>	28 secondi
<b>Punteggio</b>	1,00/1,00
<b>Valutazione</b>	<b>10,00</b> su un massimo di 10,00 ( <b>100%</b> )

**Domanda 1**

Risposta corretta

Punteggio ottenuto 1,00 su 1,00

[Domanda #2]

Sia  $f(n) = 7n^3 + 12n \log_5 n + 3 \log n + 23$ 

Selezionare tutte le risposte corrette (possono essere più di una)

Scegli una o più alternative:

- a.  $f(n) = \Omega(\log n)$  ✓
- b.  $f(n) = O(\log n)$
- c.  $f(n) = \Theta(\log n)$
- d.  $f(n) = \Theta(n \log n)$
- e.  $f(n) = O(n \log n)$
- f.  $f(n) = \Omega(n \log n)$  ✓

Risposta corretta.

Le risposte corrette sono:  $f(n) = \Omega(\log n)$ ,  $f(n) = \Omega(n \log n)$



**Iniziato** giovedì, 30 marzo 2023, 17:37

**Stato** Completato

**Terminato** giovedì, 30 marzo 2023, 17:37

**Tempo impiegato** 12 secondi

**Punteggio** 1,00/1,00

**Valutazione** 10,00 su un massimo di 10,00 (100%)

**Domanda 1**

Risposta corretta

Punteggio ottenuto 1,00 su 1,00

Quale delle seguenti affermazioni su partition è corretta?

Scegli un'alternativa:

- a. Partition ha bisogno di una struttura di appoggio su cui copiare gli elementi minori del pivot (riempiendo la struttura di appoggio a partire da sinistra) e maggiori (riempiendo a partire da destra): non esiste una sua implementazione "in place"
- b. Partition è una funzione ricorsiva
- c. Partition può essere implementata "in place", senza strutture di appoggio: è una funzione molto efficiente per questa ragione ✓

Risposta corretta.

La risposta corretta è: Partition può essere implementata "in place", senza strutture di appoggio: è una funzione molto efficiente per questa ragione



**Iniziato** martedì, 18 aprile 2023, 16:03

**Stato** Completato

**Terminato** martedì, 18 aprile 2023, 16:04

**Tempo impiegato** 46 secondi

**Punteggio** 1,00/1,00

**Valutazione** 10,00 su un massimo di 10,00 (100%)

### Domanda 1

Risposta corretta

Punteggio ottenuto 1,00 su 1,00

[Domanda #3]

Sia  $f(n) = 7n^3 + 12 n \log_5 n + 3 \log n + 23$

Selezionare tutte le risposte corrette (possono essere più di una)

Scegli una o più alternative:

- a.  $f(n) = \Theta(n^3)$  ✓
- b.  $f(n) = \Omega(n!)$
- c.  $f(n) = \Theta(n!)$
- d.  $f(n) = O(n!)$  ✓
- e.  $f(n) = \Omega(n^3)$  ✓
- f.  $f(n) = O(n^3)$  ✓

Risposta corretta.

Le risposte corrette sono:  $f(n) = O(n^3)$ ,  $f(n) = \Omega(n^3)$ ,  $f(n) = \Theta(n^3)$ ,  $f(n) = O(n!)$



**Iniziato** giovedì, 30 marzo 2023, 17:38

**Stato** Completato

**Terminato** giovedì, 30 marzo 2023, 17:39

**Tempo impiegato** 1 min. 30 secondi

**Punteggio** 6,00/6,00

**Valutazione** 10,00 su un massimo di 10,00 (100%)

### Domanda 1

Risposta corretta

Punteggio ottenuto 1,00 su 1,00

Qual è la complessità di quicksort?

Scegli un'alternativa:

- a. Theta ( n ) nel caso migliore, medio e peggiore
- b. Theta ( n log n ) nel caso migliore, medio e peggiore
- c. Theta ( n log n ) nel caso migliore e medio; Theta ( n^2 ) nel caso peggiore ✓

Risposta corretta.

La risposta corretta è: Theta ( n log n ) nel caso migliore e medio; Theta ( n^2 ) nel caso peggiore

### Domanda 2

Risposta corretta

Punteggio ottenuto 1,00 su 1,00

Come si caratterizza il caso migliore di quicksort?

Scegli un'alternativa:

- a. Il caso migliore di quicksort si ha quando ad ogni chiamata di partition, il pivot è il **mediano** degli elementi degli elementi compresi nella porzione dell'array A tra inizio e fine. Ovviamente questo caso migliore è teorico: non si può infatti calcolare l'elemento mediano senza riordinare prima gli elementi. ✓
- b. Il caso migliore di quicksort si ha quando ad ogni chiamata di partition su A tra "inizio" e "fine", il pivot è l'**elemento con indice "inizio"**. Questo permette infatti di evitare lo spostamento del pivot in prima posizione, con un notevole vantaggio in termini di efficienza.
- c. Il caso migliore di quicksort si ha quando ad ogni chiamata di partition, il pivot è la **media** degli elementi degli elementi compresi nella porzione dell'array A tra inizio e fine: calcolare il pivot richiede tempo lineare perché la media di (fine-inizio+1) elementi si calcola sommando questi elementi e dividendo per (fine-inizio+1). Calcolando la media, si può sempre ricadere nel caso migliore.

Risposta corretta.

La risposta corretta è: Il caso migliore di quicksort si ha quando ad ogni chiamata di partition, il pivot è il **mediano** degli elementi degli elementi compresi nella porzione dell'array A tra inizio e fine. Ovviamente questo caso migliore è teorico: non si può infatti calcolare l'elemento mediano senza riordinare prima gli elementi.

**Domanda 3**

Risposta corretta

Punteggio ottenuto 1,00 su 1,00

Nella complessità di quicksort nel caso migliore, Theta (  $n \log n$  ), da dove viene il fattore "n" ?

Scegli un'alternativa:

- a. Dalla complessità di partition che è Theta (  $n$  )
- b. Dal numero di livelli dell'albero della ricorsione, che è  $n$
- c. Dal numero di operazioni svolte ad ogni livello dell'albero della ricorsione: al livello  $j$  si effettuano  $2^j$  chiamate a partition, ciascuna su una porzione di array lunga  $n/(2^j)$ . Ciascuna di queste chiamate ha complessità lineare nella dimensione della porzione di array su cui viene chiamata, quindi Theta (  $n/(2^j)$  ): ad ogni livello faccio  $2^j * \Theta(n/(2^j))$  operazioni, che determinano la complessità Theta(  $n$  )



Risposta corretta.

La risposta corretta è: Dal numero di operazioni svolte ad ogni livello dell'albero della ricorsione: al livello  $j$  si effettuano  $2^j$  chiamate a partition, ciascuna su una porzione di array lunga  $n/(2^j)$ . Ciascuna di queste chiamate ha complessità lineare nella dimensione della porzione di array su cui viene chiamata, quindi Theta (  $n/(2^j)$  ): ad ogni livello faccio  $2^j * \Theta(n/(2^j))$  operazioni, che determinano la complessità Theta(  $n$  )

**Domanda 4**

Risposta corretta

Punteggio ottenuto 1,00 su 1,00

Nella complessità di quicksort nel caso migliore, Theta (  $n \log n$  ), da dove viene il fattore "log n" ?

Scegli un'alternativa:

- a. Dal costo di partition, che è Theta (  $\log n$  ) nel caso migliore
- b. Dal costo della ricerca binaria effettuata da partition per partizionare l'array in modo bilanciato
- c. Dal numero dei livelli dell'albero della ricorsione



Risposta corretta.

La risposta corretta è: Dal numero dei livelli dell'albero della ricorsione

**Domanda 5**

Risposta corretta

Punteggio ottenuto 1,00 su 1,00

Quando si ricade nel caso peggiore di quicksort?

Scegli un'alternativa:

- a. Quando l'array è ordinato in maniera decrescente e partition chiamata su A tra "inizio" e "fine" seleziona sempre come pivot **l'elemento con indice "fine"**. Questo è l'unico caso peggiore.
- b. Quando partition chiamata su A tra "inizio" e "fine" seleziona sempre come pivot **l'elemento maggiore o l'elemento minore** tra quelli compresi tra "inizio" e "fine" ✓
- c. Quando l'array è ordinato in maniera crescente e partition chiamata su A tra "inizio" e "fine" seleziona sempre come pivot **l'elemento con indice "inizio"**. Questo è l'unico caso peggiore.

Risposta corretta.

La risposta corretta è: Quando partition chiamata su A tra "inizio" e "fine" seleziona sempre come pivot **l'elemento maggiore o l'elemento minore** tra quelli compresi tra "inizio" e "fine"

**Domanda 6**

Risposta corretta

Punteggio ottenuto 1,00 su 1,00

Come si calcola la complessità nel caso peggiore di quicksort?

Scegli un'alternativa:

- a. Si sommano le operazioni fatte ad ogni livello dell'albero delle chiamate ricorsive. Tali operazioni sono dovute alla chiamata di partition, che al livello 0 verrà chiamata su n elementi (quindi n operazioni), al livello 1 verrà chiamata su n-1 elementi (quindi n-1 operazioni), al livello 2 verrà chiamata su n-2 elementi (quindi n-2 operazioni), ..., al livello n verrà chiamata su 1 elemento (una sola operazione, e poi non ci sono altre chiamate ricorsive). Questo calcolo si sviluppa nella sommatoria per i che va da 1 a n di i, in Theta(  $n^2$  ) ✓
- b. Si fa una stima per eccesso: nel caso peggiore non siamo sicuri che la complessità sia Theta(  $n \log n$  ), potrebbe essere anche peggiore, allora diciamo che è in Big-Oh(  $n^2$  ) e siamo tranquilli
- c. Si calcolano i livelli dell'albero della ricorsione, che sono n, e si moltiplica questo n per il numero di operazioni a ciascun livello, che sono Theta( n ): il risultato è Theta(  $n^2$  )

Risposta corretta.

La risposta corretta è: Si sommano le operazioni fatte ad ogni livello dell'albero delle chiamate ricorsive. Tali operazioni sono dovute alla chiamata di partition, che al livello 0 verrà chiamata su n elementi (quindi n operazioni), al livello 1 verrà chiamata su n-1 elementi (quindi n-1 operazioni), al livello 2 verrà chiamata su n-2 elementi (quindi n-2 operazioni), ..., al livello n verrà chiamata su 1 elemento (una sola operazione, e poi non ci sono altre chiamate ricorsive). Questo calcolo si sviluppa nella sommatoria per i che va da 1 a n di i, in Theta(  $n^2$  )

**Iniziato** martedì, 18 aprile 2023, 16:04

**Stato** Completato

**Terminato** martedì, 18 aprile 2023, 16:09

**Tempo impiegato** 5 min.

**Punteggio** 7,00/9,00

**Valutazione** 7,78 su un massimo di 10,00 (77,78%)

### Domanda 1

Risposta corretta

Punteggio ottenuto 1,00 su 1,00

#### Contesto:

Analisi di complessità delle operazioni su liste, Struttura dati: liste doppiamente collegate, circolari e con sentinella

Dimensione della lista: n

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

Nel contesto indicato sopra, qual è la complessità, 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

Risposta corretta.

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

**Domanda 2**

Risposta corretta

Punteggio ottenuto 1,00 su 1,00

**Contesto:**

Analisi di complessità delle operazioni su liste, Struttura dati: liste doppiamente collegate, circolari e con sentinella

Dimensione della lista: n

N indica l'insieme dei numeri naturali ed è il codominio di size

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

Scegli un'alternativa:

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

Risposta corretta.

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

**Domanda 3**

Risposta corretta

Punteggio ottenuto 1,00 su 1,00

**Contesto:**

Analisi di complessità delle operazioni su liste, Struttura dati: liste doppiamente collegate, circolari e con sentinella

Nel contesto indicato sopra, qual è la complessità, nel caso migliore e peggiore, di **emptyList:→List**

Scegli un'alternativa:

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

Risposta corretta.

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

**Domanda 4**

Risposta corretta

Punteggio ottenuto 1,00 su 1,00

**Contesto:**

Analisi di complessità delle operazioni su liste, Struttura dati: liste doppiamente collegate, circolari e con sentinella

Dimensione della lista: n

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

Nel contesto indicato sopra, qual è la complessità, nel caso migliore e peggiore, di **add: 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 e peggiore
- b. Theta ( 1 ) nel caso migliore; Theta ( n ) nel caso peggiore ✓
- c. Theta ( n ) nel caso migliore e peggiore

Risposta corretta.

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

**Domanda 5**

Risposta corretta

Punteggio ottenuto 1,00 su 1,00

**Contesto:**

Analisi di complessità delle operazioni su liste, Struttura dati: liste doppiamente collegate, circolari e con sentinella

Dimensione della lista: n

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

Nel contesto indicato sopra, qual è la complessità, nel caso migliore e peggiore, di **get: N x List → Elem***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 e peggiore
- b. Theta ( n ) nel caso migliore e peggiore
- c. Theta ( 1 ) (se la posizione di passata a get == 0) nel caso migliore; Theta ( n ) (se la posizione passata a get >= n ) nel caso peggiore ✓

Risposta corretta.

La risposta corretta è: Theta ( 1 ) (se la posizione di passata a get == 0) nel caso migliore; Theta ( n ) (se la posizione passata a get &gt;= n ) nel caso peggiore

**Domanda 6**

Risposta corretta

Punteggio ottenuto 1,00 su 1,00

**Contesto:**

Analisi di complessità delle operazioni su liste, Struttura dati: liste doppiamente collegate, circolari e con sentinella

Dimensione della lista: n

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

Scegli un'alternativa:

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

Risposta corretta.

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

**Domanda 7**

Risposta corretta

Punteggio ottenuto 1,00 su 1,00

**Contesto:**

Analisi di complessità delle operazioni su liste, Struttura dati: liste doppiamente collegate, circolari e con sentinella

Dimensione della lista: n

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

Nel contesto indicato sopra, qual è la complessità, nel caso migliore e peggiore, di **removePos: N 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 ( n ) nel caso migliore e peggiore
- b. Theta ( 1 ) nel caso migliore e peggiore
- c. Theta ( 1 ) nel caso migliore, Theta ( n ) nel caso peggiore ✓

Risposta corretta.

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

**Domanda 8**

Risposta corretta

Punteggio ottenuto 1,00 su 1,00

**Contesto:**

Analisi di complessità delle operazioni su liste, Struttura dati: liste doppiamente collegate, circolari e con sentinella

Dimensione della lista: n

Nel contesto indicato sopra, qual è la complessità, nel caso migliore e peggiore, di **isEmpty: List → Bool**

Scegli un'alternativa:

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

Risposta corretta.

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

**Domanda 9**

Risposta errata

Punteggio ottenuto -1,00 su 1,00

**Contesto:**

Analisi di complessità delle operazioni su liste, Struttura dati: liste doppiamente collegate, circolari e con sentinella

Dimensione della lista: n

Nel contesto indicato sopra, qual è la complessità, nel caso migliore e peggiore, di **addBack: Elemt x List → List**

Scegli un'alternativa:

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

Risposta errata.

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

<b>Iniziato</b>	martedì, 18 aprile 2023, 16:10
<b>Stato</b>	Completato
<b>Terminato</b>	martedì, 18 aprile 2023, 16:11
<b>Tempo impiegato</b>	1 min. 13 secondi
<b>Punteggio</b>	1,00/9,00
<b>Valutazione</b>	1,11 su un massimo di 10,00 ( <b>11,11%</b> )

**Domanda 1**

Risposta corretta

Punteggio ottenuto 1,00 su 1,00

**Contesto:**

Analisi di complessità delle operazioni su liste, Struttura dati: liste semplici

Dimensione della lista: n

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

Scegli un'alternativa:

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

Risposta corretta.

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

**Domanda 2**

Risposta corretta

Punteggio ottenuto 1,00 su 1,00

**Contesto:**

Analisi di complessità delle operazioni su liste, Struttura dati: liste semplici

Dimensione della lista: n

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

Nel contesto indicato sopra, qual è la complessità, nel caso migliore e peggiore, di **removePos: N 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 ( n ) nel caso migliore e peggiore
- c. Theta ( 1 ) nel caso migliore e peggiore

Risposta corretta.

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

**Domanda 3**

Risposta corretta

Punteggio ottenuto 1,00 su 1,00

**Contesto:**

Analisi di complessità delle operazioni su liste, Struttura dati: liste semplici

Dimensione della lista: n

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

Nel contesto indicato sopra, qual è la complessità, nel caso migliore e peggiore, di **get: N x List → Elem***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 ( n ) nel caso migliore e peggiore
- b. Theta ( 1 ) (se la posizione di passata a get == 0) nel caso migliore; Theta ( n ) (se la posizione passata a get >= n ) nel caso peggiore ✓
- c. Theta ( 1 ) nel caso migliore e peggiore

Risposta corretta.

La risposta corretta è: Theta ( 1 ) (se la posizione di passata a get == 0) nel caso migliore; Theta ( n ) (se la posizione passata a get &gt;= n ) nel caso peggiore

**Domanda 4**

Risposta corretta

Punteggio ottenuto 1,00 su 1,00

**Contesto:**

Analisi di complessità delle operazioni su liste, Struttura dati: liste semplici

Dimensione della lista: n

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

Nel contesto indicato sopra, qual è la complessità, 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( n ) nel caso migliore e peggiore
- c. Theta( 1 ) nel caso migliore e peggiore

Risposta corretta.

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

**Domanda 5**

Risposta corretta

Punteggio ottenuto 1,00 su 1,00

**Contesto:**

Analisi di complessità delle operazioni su liste, Struttura dati: liste semplici

Dimensione della lista: n

N indica l'insieme dei numeri naturali ed è il codominio di size

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

Scegli un'alternativa:

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

Risposta corretta.

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

**Domanda 6**

Risposta errata

Punteggio ottenuto -1,00 su 1,00

**Contesto:**

Analisi di complessità delle operazioni su liste, Struttura dati: liste semplici

Dimensione della lista: n

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

Nel contesto indicato sopra, qual è la complessità, nel caso migliore e peggiore, di **add: 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 e peggiore
- b. Theta ( 1 ) nel caso migliore; Theta ( n ) nel caso peggiore
- c. Theta ( n ) nel caso migliore e peggiore X

Risposta errata.

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

**Domanda 7**

Risposta errata

Punteggio ottenuto -1,00 su 1,00

**Contesto:**

Analisi di complessità delle operazioni su liste, Struttura dati: liste semplici

Dimensione della lista: n

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

Scegli un'alternativa:

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

Risposta errata.

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

**Domanda 8**

Risposta errata

Punteggio ottenuto -1,00 su 1,00

**Contesto:**

Analisi di complessità delle operazioni su liste, Struttura dati: liste semplici

Dimensione della lista: n

Nel contesto indicato sopra, qual è la complessità, nel caso migliore e peggiore, di **isEmpty: List → Bool**

Scegli un'alternativa:

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

Risposta errata.

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

**Domanda 9**

Risposta errata

Punteggio ottenuto -1,00 su 1,00

**Contesto:**

Analisi di complessità delle operazioni su liste, Struttura dati: liste semplici

Nel contesto indicato sopra, qual è la complessità, nel caso migliore e peggiore, di **emptyList:→List**

Scegli un'alternativa:

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

Risposta errata.

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

**Iniziato** martedì, 18 aprile 2023, 16:12

**Stato** Completato

**Terminato** martedì, 18 aprile 2023, 16:12

**Tempo impiegato** 52 secondi

**Punteggio** -1,00/9,00

**Valutazione** -1,11 su un massimo di 10,00 (-11,11%)

### Domanda 1

Risposta errata

Punteggio ottenuto -1,00 su 1,00

#### Contesto:

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

Dimensione della lista: n

Nel contesto indicato sopra, qual è la complessità, nel caso migliore e peggiore, di **isEmpty: List → Bool**

Scegli un'alternativa:

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

Risposta errata.

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

**Domanda 2**

Risposta corretta

Punteggio ottenuto 1,00 su 1,00

**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, 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( n ) nel caso migliore e peggiore
- b. Theta( 1 ) nel caso migliore e peggiore ✓
- c. Theta( 1 ) nel caso migliore, Theta( n ) nel caso peggiore

Risposta corretta.

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

**Domanda 3**

Risposta errata

Punteggio ottenuto -1,00 su 1,00

**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, nel caso migliore e peggiore, di **get: N x List → Elem***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 e peggiore
- b. Theta ( n ) nel caso migliore e peggiore
- c. Theta ( 1 ) (se la posizione di passata a get == 0) nel caso migliore; Theta ( n ) (se la posizione passata a get >= n ) nel caso peggiore ✗

Risposta errata.

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

**Domanda 4**

Risposta corretta

Punteggio ottenuto 1,00 su 1,00

**Contesto:**

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

Dimensione della lista: n

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

Scegli un'alternativa:

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

Risposta corretta.

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

**Domanda 5**

Risposta errata

Punteggio ottenuto -1,00 su 1,00

**Contesto:**

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

Nel contesto indicato sopra, qual è la complessità, nel caso migliore e peggiore, di **emptyList:→List**

Scegli un'alternativa:

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

Risposta errata.

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

**Domanda 6**

Risposta errata

Punteggio ottenuto -1,00 su 1,00

**Contesto:**

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

Dimensione della lista: n

N indica l'insieme dei numeri naturali ed è il codominio di size

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

Scegli un'alternativa:

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

Risposta errata.

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

**Domanda 7**

Risposta corretta

Punteggio ottenuto 1,00 su 1,00

**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, nel caso migliore e peggiore, di **add: N × Elem × 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 ) (se aggiungo in fondo e non devo ridimensionare) nel caso migliore; Theta ( n ) nel caso peggiore ✓
- b. Theta ( 1 ) nel caso migliore e peggiore
- c. Theta ( n ) nel caso migliore e peggiore

Risposta corretta.

La risposta corretta è: Theta ( 1 ) (se aggiungo in fondo e non devo ridimensionare) nel caso migliore; Theta ( n ) nel caso peggiore

**Domanda 8**

Risposta errata

Punteggio ottenuto -1,00 su 1,00

**Contesto:**

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

Dimensione della lista: n

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

Scegli un'alternativa:

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

Risposta errata.

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

**Domanda 9**

Risposta corretta

Punteggio ottenuto 1,00 su 1,00

**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, nel caso migliore e peggiore, di **removePos: N 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 e peggiore
- b. Theta ( n ) nel caso migliore e peggiore
- c. Theta ( 1 ) ( se rimuovo dalla posizione n e non devo ridimensionare ) nel caso migliore, Theta ( n ) nel caso peggiore ✓

Risposta corretta.

La risposta corretta è: Theta ( 1 ) ( se rimuovo dalla posizione n e non devo ridimensionare ) nel caso migliore, Theta ( n ) nel caso peggiore

<b>Iniziato</b>	martedì, 18 aprile 2023, 16:13
<b>Stato</b>	Completato
<b>Terminato</b>	martedì, 18 aprile 2023, 16:14
<b>Tempo impiegato</b>	1 min.
<b>Punteggio</b>	-8,00/12,00
<b>Valutazione</b>	-6,67 su un massimo di 10,00 (-66,67%)

**Domanda 1**

Risposta corretta

Punteggio ottenuto 1,00 su 1,00

**Contesto:**

Analisi di complessità delle operazioni su stack, Struttura dati: liste doppiamente collegate, circolari e con sentinella

Dimensione dello stack: n

Nel contesto indicato sopra, qual è la complessità, nel caso migliore e peggiore, di **pop**: **Stack → Elemt x Stack**

Scegli un'alternativa:

- a. Theta (n) nel caso migliore e peggiore
- b. Theta (1) nel caso migliore e peggiore ✓
- c. Theta (1) nel caso migliore; Theta (n) nel caso peggiore

Risposta corretta.

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

**Domanda 2**

Risposta errata

Punteggio ottenuto -1,00 su 1,00

**Contesto:**

Analisi di complessità delle operazioni su stack, Struttura dati: liste doppiamente collegate, circolari e con sentinella

Dimensione dello stack: n

Nel contesto indicato sopra, qual è la complessità, nel caso migliore e peggiore, di **isEmpty: Stack → Bool**

Scegli un'alternativa:

- a. Theta ( n ) nel caso migliore e peggiore
- b. Theta ( 1 ) nel caso migliore; Theta ( n ) nel caso peggiore ✗
- c. Theta ( 1 ) nel caso migliore e peggiore

Risposta errata.

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

**Domanda 3**

Risposta errata

Punteggio ottenuto -1,00 su 1,00

**Contesto:**

Analisi di complessità delle operazioni su stack, Struttura dati: liste semplici

Dimensione dello stack: n

Nel contesto indicato sopra, qual è la complessità, nel caso migliore e peggiore, di **push: Elemt x Stack → Stack****Attenzione:** pensare a come implementare push e pop in modo da renderle il più efficienti possibile, non soffermarsi sull'idea di "ultimo" elemento di una lista

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

Risposta errata.

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

**Domanda 4**

Risposta errata

Punteggio ottenuto -1,00 su 1,00

**Contesto:**

Analisi di complessità delle operazioni su stack, Struttura dati: array dinamico con size, maxsize e ridimensionamento

Dimensione dello stack: n

Nel contesto indicato sopra, qual è la complessità, nel caso migliore e peggiore, di **isEmpty: Stack → Bool***Attenzione: pensare a come implementare push e pop in modo da renderle il più efficienti possibile*

Scegli un'alternativa:

- a. Theta ( 1 ) nel caso migliore; Theta ( n ) nel caso peggiore
- b. Theta ( n ) nel caso migliore e peggiore **✗**
- c. Theta ( 1 ) nel caso migliore e peggiore

Risposta errata.

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

**Domanda 5**

Risposta errata

Punteggio ottenuto -1,00 su 1,00

**Contesto:**

Analisi di complessità delle operazioni su stack, Struttura dati: liste collegate semplicemente

Dimensione dello stack: n

Nel contesto indicato sopra, qual è la complessità, nel caso migliore e peggiore, di **isEmpty: Stack → Bool***Attenzione: pensare a come implementare push e pop in modo da renderle il più efficienti possibile, non soffermarsi sull'idea di "ultimo" elemento di una lista*

Scegli un'alternativa:

- a. Theta ( n ) nel caso migliore e peggiore **✗**
- b. Theta ( 1 ) nel caso migliore e peggiore
- c. Theta ( 1 ) nel caso migliore; Theta ( n ) nel caso peggiore

Risposta errata.

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

**Domanda 6**

Risposta errata

Punteggio ottenuto -1,00 su 1,00

**Contesto:**

Analisi di complessità delle operazioni su stack, Struttura dati: liste doppiamente collegate, circolari e con sentinella

Dimensione dello stack: n

Nel contesto indicato sopra, qual è la complessità, nel caso migliore e peggiore, di **push: Elem x Stack → Stack**

Scegli un'alternativa:

- a. Theta ( 1 ) nel caso migliore e peggiore
- b. Theta ( n ) nel caso migliore e peggiore
- c. Theta ( 1 ) nel caso migliore; Theta ( n ) nel caso peggiore X

Risposta errata.

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

**Domanda 7**

Risposta errata

Punteggio ottenuto -1,00 su 1,00

**Contesto:**

Analisi di complessità delle operazioni su stack, Struttura dati: liste semplici

Dimensione dello stack: n

Nel contesto indicato sopra, qual è la complessità, nel caso migliore e peggiore, di **pop: Stack → Elem x Stack***Attenzione: pensare a come implementare push e pop in modo da renderle il più efficienti possibile, non soffermarsi sull'idea di "ultimo" elemento di una lista*

Scegli un'alternativa:

- a. Theta ( n ) nel caso migliore e peggiore
- b. Theta ( 1 ) nel caso migliore e peggiore
- c. Theta ( 1 ) nel caso migliore; Theta ( n ) nel caso peggiore X

Risposta errata.

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

**Domanda 8**

Risposta errata

Punteggio ottenuto -1,00 su 1,00

**Contesto:**

Analisi di complessità delle operazioni su stack, Struttura dati: array dinamico con size, maxsize e ridimensionamento

Dimensione dello stack: n

Nel contesto indicato sopra, qual è la complessità, nel caso migliore e peggiore, di **top: Stack → Elem x Stack***Attenzione: pensare a come implementare push e pop in modo da renderle il più efficienti possibile*

Scegli un'alternativa:

- a. Theta ( n ) nel caso migliore e peggiore
- b. Theta ( 1 ) nel caso migliore; Theta ( n ) nel caso peggiore ✗
- c. Theta ( 1 ) nel caso migliore e peggiore

Risposta errata.

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

**Domanda 9**

Risposta errata

Punteggio ottenuto -1,00 su 1,00

**Contesto:**

Analisi di complessità delle operazioni su stack, Struttura dati: liste doppiamente collegate, circolari e con sentinella

Dimensione dello stack: n

Nel contesto indicato sopra, qual è la complessità, nel caso migliore e peggiore, di **top: Stack → Elem x Stack**

Scegli un'alternativa:

- a. Theta ( 1 ) nel caso migliore e peggiore
- b. Theta ( n ) nel caso migliore e peggiore
- c. Theta ( 1 ) nel caso migliore; Theta ( n ) nel caso peggiore ✗

Risposta errata.

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

**Domanda 10**

Risposta errata

Punteggio ottenuto -1,00 su 1,00

**Contesto:**

Analisi di complessità delle operazioni su stack, Struttura dati: array dinamico con size, maxsize e ridimensionamento

Dimensione dello stack: n

Nel contesto indicato sopra, qual è la complessità, nel caso migliore e peggiore, di **pop: Stack → Eleml x Stack***Attenzione: pensare a come implementare push e pop in modo da renderle il più efficienti possibile*

Scegli un'alternativa:

- a. Theta ( n ) nel caso migliore e peggiore
- b. Theta ( 1 ) nel caso migliore e peggiore ✗
- c. Theta ( 1 ) nel caso migliore; Theta ( n ) nel caso peggiore, se si deve ridimensionare

Risposta errata.

La risposta corretta è: Theta ( 1 ) nel caso migliore; Theta ( n ) nel caso peggiore, se si deve ridimensionare

**Domanda 11**

Risposta errata

Punteggio ottenuto -1,00 su 1,00

**Contesto:**

Analisi di complessità delle operazioni su stack, Struttura dati: liste semplici

Dimensione dello stack: n

Nel contesto indicato sopra, qual è la complessità, nel caso migliore e peggiore, di **top: Stack → Eleml x Stack***Attenzione: pensare a come implementare push e pop in modo da renderle il più efficienti possibile, non soffermarsi sull'idea di "ultimo" elemento di una lista*

Scegli un'alternativa:

- a. Theta ( 1 ) nel caso migliore; Theta ( n ) nel caso peggiore
- b. Theta ( n ) nel caso migliore e peggiore ✗
- c. Theta ( 1 ) nel caso migliore e peggiore

Risposta errata.

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

**Domanda 12**

Risposta corretta

Punteggio ottenuto 1,00 su 1,00

**Contesto:**

Analisi di complessità delle operazioni su stack, Struttura dati: array dinamico con size, maxsize e ridimensionamento

Dimensione dello stack: n

Nel contesto indicato sopra, qual è la complessità, nel caso migliore e peggiore, di **push: Elem x Stack → Stack***Attenzione: pensare a come implementare push e pop in modo da renderle il più efficienti possibile*

Scegli un'alternativa:

- a. Theta (n) nel caso migliore e peggiore
- b. Theta (1) nel caso migliore e peggiore
- c. Theta (1) nel caso migliore; Theta (n) nel caso peggiore in cui devo ridimensionare ✓

Risposta corretta.

La risposta corretta è: Theta (1) nel caso migliore; Theta (n) nel caso peggiore in cui devo ridimensionare

<b>Iniziato</b>	martedì, 18 aprile 2023, 16:14
<b>Stato</b>	Completato
<b>Terminato</b>	martedì, 18 aprile 2023, 16:15
<b>Tempo impiegato</b>	31 secondi
<b>Punteggio</b>	0,00/2,00
<b>Valutazione</b>	0,00 su un massimo di 10,00 (0%)

**Domanda 1**

Risposta corretta

Punteggio ottenuto 1,00 su 1,00

Qual è il modo più efficiente di progettare le operazioni sulle code, quando implementate come array dinamico con size, maxsize e ridimensionamento?

Scegli un'alternativa:

- a. Non importa dove mette l'elemento enqueue: infatti dequeue e first hanno sempre complessità theta ( 1 ) nel caso migliore e theta ( n ) nel caso peggiore
- b. o rendo efficiente la enqueue, o rendo efficiente la dequeue: nel caso migliore, una delle due ha complessità Theta ( 1 ) ma l'altra ha complessità Theta ( n ) ✓
- c. posso rendere efficiente sia la enqueue che la dequeue: con una progettazione furba, nel caso migliore entrambe hanno complessità Theta ( 1 )

Risposta corretta.

La risposta corretta è: o rendo efficiente la enqueue, o rendo efficiente la dequeue: nel caso migliore, una delle due ha complessità Theta ( 1 ) ma l'altra ha complessità Theta ( n )

**Domanda 2**

Risposta errata

Punteggio ottenuto -1,00 su 1,00

Qual è il modo più efficiente di progettare le operazioni sulle code, quando implementate come liste semplici?

Scegli un'alternativa:

- a. Se enqueue mette l'elemento in fondo alla lista la sua complessità è Theta( n ) nel caso migliore e peggiore, ma dequeue e first hanno sempre complessità Theta ( 1 )
- b. Se enqueue mette l'elemento all'inizio della lista semplice la sua complessità è Theta( n ) nel caso migliore e peggiore, ma dequeue e first hanno sempre complessità Theta ( 1 ) ✗
- c. Non importa dove mette l'elemento enqueue: infatti dequeue e first hanno sempre complessità theta ( 1 ) nel caso migliore e theta ( n ) nel caso peggiore

Risposta errata.

La risposta corretta è: Se enqueue mette l'elemento in fondo alla lista la sua complessità è Theta( n ) nel caso migliore e peggiore, ma dequeue e first hanno sempre complessità Theta ( 1 )

**Iniziato** martedì, 18 aprile 2023, 16:15

**Stato** Completato

**Terminato** martedì, 18 aprile 2023, 16:16

**Tempo impiegato** 31 secondi

**Punteggio** -4,00/6,00

**Valutazione** -6,67 su un massimo di 10,00 (-66,67%)

### Domanda 1

Risposta errata

Punteggio ottenuto -1,00 su 1,00

#### Contesto:

Analisi di complessità delle operazioni su set, Struttura dati: array **ORDINATO** con size e maxsize, assumendo per semplicità che non sia necessario ridimensionare

Dimensione del primo insieme: **n**

Dimensione del secondo insieme: **m**

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

Scegli un'alternativa:

- a. Se gli array che rappresentano i set sono ORDINATI, e assumendo che non si debba ridimensionare, la complessità di setUnion è Theta ( $n + m$ ) nel caso migliore e peggiore, usando una tecnica simile a quella di merge e sfruttando l'informazione che gli array sono ordinati
- b. Se gli array che rappresentano i set sono ORDINATI, e assumendo che non si debba ridimensionare, la complessità di setUnion è Theta ( $\min(n, m, n*m)$ ) nel caso migliore e peggiore, usando una tecnica simile a quella di merge e sfruttando l'informazione che gli array sono ordinati ✗
- c. La complessità non dipende dal fatto che gli array siano ordinati ed è Theta ( $n*m$ ) nel caso migliore e peggiore

Risposta errata.

La risposta corretta è: Se gli array che rappresentano i set sono ORDINATI, e assumendo che non si debba ridimensionare, la complessità di setUnion è Theta ( $n + m$ ) nel caso migliore e peggiore, usando una tecnica simile a quella di merge e sfruttando l'informazione che gli array sono ordinati

**Domanda 2**

Risposta corretta

Punteggio ottenuto 1,00 su 1,00

**Contesto:**

Analisi di complessità delle operazioni su set, Struttura dati: liste doppiamente collegate, circolari e con sentinella

Dimensione del set: **n**Nel contesto indicato sopra, qual è la complessità, nel caso migliore e peggiore, di **insertElem: Elem x Set → Set**

Scegli un'alternativa:

- a. Theta ( 1 ) nel caso migliore e peggiore
- b. Theta ( n ) nel caso migliore e peggiore
- c. Theta ( 1 ) nel caso migliore; Theta ( n ) nel caso peggiore ✓

Risposta corretta.

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

**Domanda 3**

Risposta errata

Punteggio ottenuto -1,00 su 1,00

**Contesto:**

Analisi di complessità delle operazioni su set, Struttura dati: liste doppiamente collegate, circolari e con sentinella

Dimensione del primo insieme: **n**Dimensione del secondo insieme: **m**Nel contesto indicato sopra, qual è la complessità, nel caso migliore e peggiore, di **setUnion: Set x Set → Set**

Scegli un'alternativa:

- a. Theta ( max( n, m, n\*m ) ) nel caso migliore e peggiore
- b. Theta ( min( n, m, n\*m ) ) nel caso migliore e peggiore ✗
- c. Theta ( n\*m ) nel caso migliore e peggiore

Risposta errata.

La risposta corretta è: Theta ( max( n, m, n\*m ) ) nel caso migliore e peggiore

**Domanda 4**

Risposta errata

Punteggio ottenuto -1,00 su 1,00

**Contesto:**

Analisi di complessità delle operazioni su set, Struttura dati: array dinamici con size, maxsize e ridimensionamento

Dimensione del primo insieme: **n**Dimensione del secondo insieme: **m**Nel contesto indicato sopra, qual è la complessità, nel caso migliore e peggiore, di **setIntersection: Set x Set → Set**

Scegli un'alternativa:

- a. Theta ( min( n, m, n\*m ) ) nel caso migliore e peggiore
- b. Theta ( n\*m ) nel caso migliore e peggiore
- c. Theta ( max( n, m, n\*m ) ) nel caso migliore e peggiore **X**

Risposta errata.

La risposta corretta è: Theta ( n\*m ) nel caso migliore e peggiore

**Domanda 5**

Risposta errata

Punteggio ottenuto -1,00 su 1,00

**Contesto:**

Analisi di complessità delle operazioni su set, Struttura dati: bit vector

Dimensione (fissa e stabilità a priori) del bit vector: **n**Nel contesto indicato sopra, qual è la complessità, nel caso migliore e peggiore, di **emptySet: → Set**

Scegli un'alternativa:

- a. Theta ( 1 ) nel caso migliore e peggiore
- b. Theta ( 1 ) nel caso migliore e Theta ( n ) nel caso peggiore **X**
- c. Theta ( n ) nel caso migliore e peggiore

Risposta errata.

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

**Domanda 6**

Risposta errata

Punteggio ottenuto -1,00 su 1,00

**Contesto:**

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

Dimensione del set: n

Nel contesto indicato sopra, qual è la complessità, nel caso migliore e peggiore, di **insertElem: Elem x Set → Set**

Scegli un'alternativa:

- a. Theta ( 1 ) nel caso migliore e peggiore
- b. Theta ( n ) nel caso migliore e peggiore ✗
- c. Theta ( 1 ) nel caso migliore; Theta ( n ) nel caso peggiore

Risposta errata.

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

<b>Iniziato</b>	lunedì, 1 maggio 2023, 11:31
<b>Stato</b>	Completato
<b>Terminato</b>	lunedì, 1 maggio 2023, 11:34
<b>Tempo impiegato</b>	3 min. 13 secondi
<b>Punteggio</b>	2,00/4,00
<b>Valutazione</b>	5,00 su un massimo di 10,00 (50%)

**Domanda 1**

Risposta corretta

Punteggio ottenuto 1,00 su 1,00

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.

Risposta corretta.

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 errata

Punteggio ottenuto -1,00 su 1,00

Si consideri un dizionario in cui le chiavi sono stringhe s di due caratteri; indichiamo tali caratteri con s[0] ed s[1] rispettivamente. In C++ il codice ascii del carattere s[i] è dato da s[i] stesso; i valori degli elementi del dizionario non interessano ai fini dell'esercizio.

Il dizionario viene implementato mediante una tabella di hash v di dimensione m (indici ammissibili: [0, m-1]).

Quali delle seguenti funzioni di hash è corretta per l'implementazione del dizionario con la tabella v?

Scegli un'alternativa:

- a.  $h(s) = s[0]$  ❌ Sbagliato: s[0] - s[1] potrebbe essere un numero  $\geq 0$  oppure  $< 0$ , e quindi non essere un indice corretto per la tabella di hash v, che ha dimensione m
- b.  $h(s) = (s[0] + s[1]) \bmod m$
- c.  $h(s) = s[0] + s[1]$

Risposta errata.

La risposta corretta è:  $h(s) = (s[0] + s[1]) \bmod m$

**Domanda 3**

Risposta corretta

Punteggio ottenuto 1,00 su 1,00

In una tabella di hash, il requisito principale sulla funzione di hash  $h$  è che

Scegli un'alternativa:

- a.  $h$  sia calcolabile in tempo costante ✓
- b.  $h$  sia calcolabile in tempo  $\theta(n)$ , con  $n$  numero di elementi nel dizionario implementato dalla tabella di hash
- c.  $h$  sia iniettiva

Risposta corretta.

La risposta corretta è:  $h$  sia calcolabile in tempo costante

**Domanda 4**

Risposta corretta

Punteggio ottenuto 1,00 su 1,00

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.  $\alpha = n/m$  ✓
- b.  $\alpha = m/n$
- c.  $\alpha = v[h(k)]$

Risposta corretta.

La risposta corretta è:  $\alpha = n/m$

<b>Iniziato</b>	lunedì, 1 maggio 2023, 11:35
<b>Stato</b>	Completato
<b>Terminato</b>	lunedì, 1 maggio 2023, 11:37
<b>Tempo impiegato</b>	2 min. 12 secondi
<b>Punteggio</b>	1,00/3,00
<b>Valutazione</b>	3,33 su un massimo di 10,00 (33,33%)

**Domanda 1**

Risposta corretta

Punteggio ottenuto 1,00 su 1,00

Nella struttura dati "tabella di hash con liste di collisione", oltre ad essere calcolabile in tempo costante una buona funzione di hash dovrebbe essere...

Scegli un'alternativa:

- a. in grado di **distribuire in modo uniforme** le chiavi nello spazio degli indici della tabella ✓
- b. **bigettiva**
- c. **iniettiva**
- d. **surgettiva**

Risposta corretta.

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

**Domanda 2**

Risposta corretta

Punteggio ottenuto 1,00 su 1,00

In una tabella di hash **v** con liste di collisione e funzione di hash **h** che implementa il TDD Dizionario...

Scegli un'alternativa:

- a. gli elementi sono contenuti nella tabella stessa: la coppia (k, value) appartenente al Dizionario si memorizza ponendo  $v[h(k)] = \text{value}$
- b. gli elementi sono contenuti nella tabella stessa: la coppia (k, value) appartenente al Dizionario si memorizza ponendo  $v[(h(k) \bmod m)] = \text{value}$ ; calcolare "mod m" dopo aver calcolato la funzione di hash **h** applicata alla chiave k garantisce che l'indice resti nei limiti della tabella
- c. gli elementi sono contenuti in liste esterne alla tabella (chiamate bucket):  $v[i]$  punta alla lista semplice delle coppie (k, value) appartenenti al Dizionario, tali che  $h(k)=i$  ✓

Risposta corretta.

La risposta corretta è: gli elementi sono contenuti in liste esterne alla tabella (chiamate bucket):  $v[i]$  punta alla lista semplice delle coppie (k, value) appartenenti al Dizionario, tali che  $h(k)=i$

**Domanda 3**

Risposta errata

Punteggio ottenuto -1,00 su 1,00

In una tabella di hash  $v$  che implementa il TDD Dizionario con liste di collisione e funzione di hash  $h$  con uniformità semplice e calcolabile in Theta (1), la complessità dell'operazione **insert( $k$ ,  $value$ )** è...

Scegli un'alternativa:

- a. **Theta (1) sia nel caso migliore che in quello peggiore:** inserire un elemento in testa ad una lista semplice richiede infatti solo di "aggiustare i puntatori" relativi al primo elemento della lista e al suo successore, e questo ha costo costante.
- b. **Theta (1) nel caso migliore** (il primo elemento della lista in  $v[h(k)]$  ha chiave  $k$ , quindi non devo re-inserire la nuova coppia ( $k$ ,  $value$ ) oppure la sovrascrivo; in ogni caso non devo scandire tutta la lista di collisione); **Theta (1 + n/m) nel caso peggiore** (nella lista di collisione in posizione  $v[h(k)]$  ci sono già tutti gli  $n/m$  elementi che devono finire in quella lista, e che sono "circa"  $n/m$  grazie al fatto che  $h$  "ha distribuito bene le chiavi rispetto agli indici"; prima di inserire la coppia ( $k$ ,  $value$ ) devo verificare se l'elemento con chiave  $k$  c'è già: se c'è ed è l'ultimo, e la lista era lunga  $n/m$ , si sono scanditi  $n/m$  elementi)
- c. **Theta (1) nel caso migliore** (il primo elemento della lista in  $v[h(k)]$  ha chiave  $k$ , quindi non devo re-inserire la nuova coppia ( $k$ ,  $value$ ) oppure la sovrascrivo; in ogni caso non devo scandire tutta la lista di collisione); **Theta (n) nel caso peggiore** (nella lista di collisione in posizione  $v[h(k)]$  si sono affollati tutti gli  $n$  elementi del dizionario; prima di fare l'inserimento si deve verificare se l'elemento con chiave  $k$  c'è già: se c'è ed è l'ultimo, e la lista era lunga  $n$ , si sono scanditi  $n$  elementi)

Risposta errata.

La risposta corretta è: **Theta (1) nel caso migliore** (il primo elemento della lista in  $v[h(k)]$  ha chiave  $k$ , quindi non devo re-inserire la nuova coppia ( $k$ ,  $value$ ) oppure la sovrascrivo; in ogni caso non devo scandire tutta la lista di collisione); **Theta (1 + n/m) nel caso peggiore** (nella lista di collisione in posizione  $v[h(k)]$  ci sono già tutti gli  $n/m$  elementi che devono finire in quella lista, e che sono "circa"  $n/m$  grazie al fatto che  $h$  "ha distribuito bene le chiavi rispetto agli indici"; prima di inserire la coppia ( $k$ ,  $value$ ) devo verificare se l'elemento con chiave  $k$  c'è già: se c'è ed è l'ultimo, e la lista era lunga  $n/m$ , si sono scanditi  $n/m$  elementi)

**Iniziato** lunedì, 1 maggio 2023, 11:38

**Stato** Completato

**Terminato** lunedì, 1 maggio 2023, 11:40

**Tempo impiegato** 2 min. 15 secondi

**Punteggio** 8,00/8,00

**Valutazione** 10,00 su un massimo di 10,00 (100%)

#### Domanda 1

Risposta corretta

Punteggio ottenuto 1,00 su 1,00

In un albero radicato ( $N, A$ ), con  $N$  insieme dei nodi (o "vertici") e  $A$  (sottoinsieme di  $N \times N$ ) insieme degli archi...

Scegli un'alternativa:

- a. ogni nodo  $v$  appartenente a  $N$  tranne la radice ha uno ed un solo figlio  $f$  tale che  $(v, f)$  appartiene ad  $A$ .
- b. ogni nodo  $v$  appartenente a  $N$  tranne la radice ha uno ed un solo genitore  $u$  tale che  $(u, v)$  appartiene ad  $A$ . ✓
- c. ogni nodo  $v$  appartenente a  $N$  ha uno ed un solo genitore  $u$  tale che  $(u, v)$  appartiene ad  $A$ .

Risposta corretta.

La risposta corretta è: ogni nodo  $v$  appartenente a  $N$  tranne la radice ha uno ed un solo genitore  $u$  tale che  $(u, v)$  appartiene ad  $A$ .

#### Domanda 2

Risposta corretta

Punteggio ottenuto 1,00 su 1,00

In un albero radicato ( $N, A$ ), con  $N$  insieme dei nodi (o "vertici") e  $A$  (sottoinsieme di  $N \times N$ ) insieme degli archi...

Scegli un'alternativa:

- a. il **grado** di un nodo  $v$  è il livello a cui il nodo si trova nell'albero
- b. il **grado** di un nodo  $v$  è il numero di figli del nodo ✓
- c. il **grado** di un nodo  $v$  è il numero di genitori del nodo

Risposta corretta.

La risposta corretta è: il **grado** di un nodo  $v$  è il numero di figli del nodo

**Domanda 3**

Risposta corretta

Punteggio ottenuto 1,00 su 1,00

In un albero radicato ( $N, A$ ), con  $N$  insieme dei nodi (o "vertici") e  $A$  (sottoinsieme di  $N \times N$ ) insieme degli archi...

Scegli un'alternativa:

- a. Un nodo senza figli è chiamato **foglia** ✓
- b. Un nodo senza figli è chiamato **radice**
- c. Un nodo senza figli è chiamato **nodo interno**

Risposta corretta.

La risposta corretta è: Un nodo senza figli è chiamato **foglia**

**Domanda 4**

Risposta corretta

Punteggio ottenuto 1,00 su 1,00

In un albero radicato ( $N, A$ ), con  $N$  insieme dei nodi (o "vertici") e  $A$  (sottoinsieme di  $N \times N$ ) insieme degli archi...

Scegli un'alternativa:

- a. Gli **antenati** di un nodo  $u$  sono tutti i nodi raggiungibili da  $u$  salendo di padre in padre nell'albero, fino a che non si raggiunge la **radice** ✓
- b. Gli **antenati** di un nodo  $u$  sono tutti i nodi allo stesso livello di  $u$  nell'albero
- c. La radice è l'unico nodo dell'albero ad avere **n-1 antenati**, dove  $n$  è il numero di nodi dell'albero

Risposta corretta.

La risposta corretta è: Gli **antenati** di un nodo  $u$  sono tutti i nodi raggiungibili da  $u$  salendo di padre in padre nell'albero, fino a che non si raggiunge la radice

**Domanda 5**

Risposta corretta

Punteggio ottenuto 1,00 su 1,00

In un albero radicato ( $N, A$ ), con  $N$  insieme dei nodi (o "vertici") e  $A$  (sottoinsieme di  $N \times N$ ) insieme degli archi...

Scegli un'alternativa:

- a. La radice è l'unico nodo dell'albero ad avere **n-1 descendenti**, dove **n** è il numero di nodi dell'albero ✓
- b. I **descendenti** di un nodo **u** sono tutti i nodi raggiungibili da **u** salendo di padre in padre nell'albero, fino a che non si raggiunge la radice
- c. I **descendenti** di un nodo **u** sono tutte e sole le foglie raggiungibili da **u** scendendo di figlio in figlio nell'albero

Risposta corretta.

La risposta corretta è: La radice è l'unico nodo dell'albero ad avere **n-1 descendenti**, dove **n** è il numero di nodi dell'albero

**Domanda 6**

Risposta corretta

Punteggio ottenuto 1,00 su 1,00

In un albero radicato ( $N, A$ ), con  $N$  insieme dei nodi (o "vertici") e  $A$  (sottoinsieme di  $N \times N$ ) insieme degli archi...

Scegli un'alternativa:

- a. La profondità (o livello) di un nodo **u** è il numero di **nodi** che bisogna attraversare per raggiungere **u** a partire dalla radice: **la radice ha quindi livello 1** perché si attraversa la radice stessa per raggiungerla
- b. La profondità (o livello) di un nodo **u** è il numero di **archi** che bisogna attraversare per raggiungere **u** a partire dalla radice: **la radice ha quindi livello n**, con **n** numero dei nodi dell'albero
- c. La profondità (o livello) di un nodo **u** è il numero di **archi** che bisogna attraversare per raggiungere **u** a partire dalla radice: **la radice ha quindi livello 0** perché non si attraversa nessun arco per raggiungerla, partendo dalla radice stessa ✓

Risposta corretta.

La risposta corretta è: La profondità (o livello) di un nodo **u** è il numero di **archi** che bisogna attraversare per raggiungere **u** a partire dalla radice: **la radice ha quindi livello 0** perché non si attraversa nessun arco per raggiungerla, partendo dalla radice stessa

**Domanda 7**

Risposta corretta

Punteggio ottenuto 1,00 su 1,00

In un albero radicato ( $N, A$ ), con  $N$  insieme dei nodi (o "vertici") e  $A$  (sottoinsieme di  $N \times N$ ) insieme degli archi...

Scegli un'alternativa:

- a. L'altezza di un albero è il grado della radice
- b. L'altezza di un albero è il numero di foglie totali
- c. L'altezza di un albero è la massima profondità a cui si trova una foglia ✓

Risposta corretta.

La risposta corretta è: L'altezza di un albero è la massima profondità a cui si trova una foglia

**Domanda 8**

Risposta corretta

Punteggio ottenuto 1,00 su 1,00

In un albero radicato ( $N, A$ ), con  $N$  insieme dei nodi (o "vertici") e  $A$  (sottoinsieme di  $N \times N$ ) insieme degli archi...

Scegli un'alternativa:

- a. La radice ha **n-1** fratelli, con **n** numero di nodi nell'albero
- b. La radice ha **0** fratelli ✓
- c. La radice ha **uno ed un solo** fratello

Risposta corretta.

La risposta corretta è: La radice ha **0** fratelli

**Iniziato** mercoledì, 31 maggio 2023, 13:31

**Stato** Completato

**Terminato** mercoledì, 31 maggio 2023, 13:34

**Tempo impiegato** 2 min. 30 secondi

**Punteggio** 0,00/8,00

**Valutazione** 0,00 su un massimo di 10,00 (0%)

#### Domanda 1

Risposta corretta

Punteggio ottenuto 1,00 su 1,00

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

Scegli un'alternativa:

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

Risposta corretta.

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

#### Domanda 2

Risposta errata

Punteggio ottenuto -1,00 su 1,00

Un albero **binario completo** è un albero in cui....

Scegli un'alternativa:

- a. tutti i nodi **tranne la radice** hanno **esattamente grado 2** ✗
- b. tutti i nodi **tranne le foglie** hanno **esattamente grado 2**
- c. i nodi hanno **al più grado 2**

Risposta errata.

La risposta corretta è: tutti i nodi **tranne le foglie** hanno **esattamente grado 2**

**Domanda 3**

Risposta corretta

Punteggio ottenuto 1,00 su 1,00

In un albero **binario completo** con altezza  $h$ ...

Scegli un'alternativa:

- a. ci sono  $2^{h+1}-1$  nodi ✓
- b. ci sono  $h$  nodi
- c. ci sono  $\log_2 h$  nodi

Risposta corretta.

La risposta corretta è: ci sono  $2^{h+1}-1$  nodi

**Domanda 4**

Risposta corretta

Punteggio ottenuto 1,00 su 1,00

In un albero **binario quasi completo** con  $n$  nodi...

Scegli un'alternativa:

- a. l'altezza è  $\lfloor \log_2 n \rfloor$  (si ricorda che  $\lfloor \cdot \rfloor$  indica la parte intera) ✓
- b. l'altezza è al più  $n^2$ , nel caso peggiore
- c. l'altezza è al più  $n$ , quando l'albero è completamente sbilanciato

Risposta corretta.

La risposta corretta è: l'altezza è  $\lfloor \log_2 n \rfloor$  (si ricorda che  $\lfloor \cdot \rfloor$  indica la parte intera)

**Domanda 5**

Risposta errata

Punteggio ottenuto -1,00 su 1,00

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

Scegli un'alternativa:

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

Risposta errata.

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

**Domanda 6**

Risposta errata

Punteggio ottenuto -1,00 su 1,00

Un BST è un...

Scegli un'alternativa:

- a. albero binario completo o quasi completo che soddisfa le seguenti proprietà: ✖
1. ogni nodo  $v$  contiene un elemento  $\text{elem}(v)$  cui è associata una chiave  $\text{chiave}(v)$  presa da un dominio totalmente ordinato
  2. le chiavi degli elementi nel sottoalbero sinistro di  $v$  sono  $\leq \text{chiave}(v)$
  3. le chiavi degli elementi nel sottoalbero destro di  $v$  sono  $\geq \text{chiave}(v)$
- b. albero binario che soddisfa le seguenti proprietà:
1. ogni nodo  $v$  contiene un elemento  $\text{elem}(v)$  cui è associata una chiave  $\text{chiave}(v)$  presa da un dominio totalmente ordinato
  2. le chiavi degli elementi a sinistra della radice sono  $\leq \text{chiave}(\text{radice})$
  3. le chiavi degli elementi a destra della radice sono  $\geq \text{chiave}(\text{radice})$
- c. albero binario che soddisfa le seguenti proprietà:
1. ogni nodo  $v$  contiene un elemento  $\text{elem}(v)$  cui è associata una chiave  $\text{chiave}(v)$  presa da un dominio totalmente ordinato
  2. le chiavi degli elementi nel sottoalbero sinistro di  $v$  sono  $\leq \text{chiave}(v)$
  3. le chiavi degli elementi nel sottoalbero destro di  $v$  sono  $\geq \text{chiave}(v)$

Risposta errata.

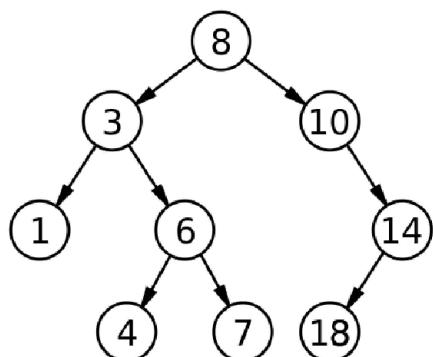
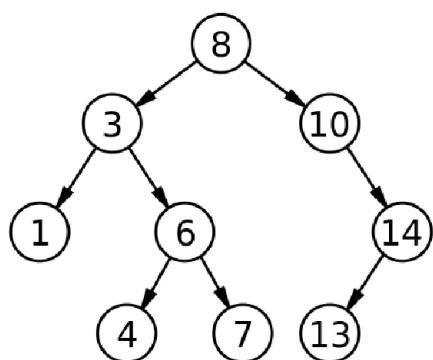
La risposta corretta è: albero binario che soddisfa le seguenti proprietà:

1. ogni nodo  $v$  contiene un elemento  $\text{elem}(v)$  cui è associata una chiave  $\text{chiave}(v)$  presa da un dominio totalmente ordinato
2. le chiavi degli elementi nel sottoalbero sinistro di  $v$  sono  $\leq \text{chiave}(v)$
3. le chiavi degli elementi nel sottoalbero destro di  $v$  sono  $\geq \text{chiave}(v)$

**Domanda 7****Risposta errata**

Punteggio ottenuto -1,00 su 1,00

Si considerino i due alberi disegnati sotto, indicati da **A** (l'albero più in alto) e **B** (l'albero più in basso):

**(A)****(B)**

Una sola delle seguenti affermazioni è corretta. Quale?

Scegli un'alternativa:

- a. **A** è un BST, mentre **B** non lo è
- b. **B** è un BST, mentre **A** non lo è
- c. Nessuno dei due alberi è un BST perché non sono **alberi binari completi** ✗
- d. Entrambi gli alberi sono BST

**Risposta errata.**

La risposta corretta è: **B** è un BST, mentre **A** non lo è

**Domanda 8**

Risposta corretta

Punteggio ottenuto 1,00 su 1,00

Si consideri la seguente struct in C++

```
struct dict::bstNode
{
    dictionaryElem keyVal; /* elemento del dictionary */
    bstNode* leftChild;
    bstNode* rightChild;
};
```

Una sola delle seguenti affermazioni è corretta. Quale?

Scegli un'alternativa:

- a. questa è la struct che useremo per implementare un nodo di un BST; manca volutamente il puntatore al padre del nodo e questo ci costringerà a progettare gli algoritmi con più cura; ne risulteranno algoritmi ricorsivi particolarmente eleganti, nei quali il puntatore al padre non serve
- b. questa struct non può essere usata per implementare un nodo di un BST perché manca il puntatore al padre del nodo: è scientificamente dimostrato che non è possibile implementare gli algoritmi di inserimento, ricerca e cancellazione del TDD Dizionario implementato mediante BST senza il puntatore al padre
- c. questa struct è adatta alla implementazione di un nodo di un albero generico, in cui il grado può essere qualunque

Risposta corretta.

La risposta corretta è: questa è la struct che useremo per implementare un nodo di un BST; manca volutamente il puntatore al padre del nodo e questo ci costringerà a progettare gli algoritmi con più cura; ne risulteranno algoritmi ricorsivi particolarmente eleganti, nei quali il puntatore al padre non serve

<b>Iniziato</b>	mercoledì, 31 maggio 2023, 13:35
<b>Stato</b>	Completato
<b>Terminato</b>	mercoledì, 31 maggio 2023, 13:35
<b>Tempo impiegato</b>	41 secondi
<b>Punteggio</b>	-4,00/8,00
<b>Valutazione</b>	-5,00 su un massimo di 10,00 (-50%)

**Domanda 1**

Risposta errata

Punteggio ottenuto -1,00 su 1,00

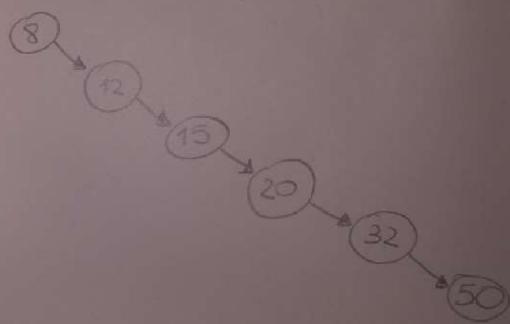
Si consideri un BST **b** = **EMPTYBST**, ovvero il BST vuoto. Qual è il risultato delle seguenti chiamate a insert su **b** (indichiamo solo la chiave e trascuriamo il valore, rappresentato da puntini), eseguite nell'ordine dall'alto verso il basso in cui sono mostrate (la prima chiamata è insert ( 8, ...); la seconda chiamata è insert ( 12, ...); ...) ?

```
insert( 8, ....);
insert( 12, ....);
insert( 15, ....);
insert( 20, ....);
insert( 32, ....);
insert( 50, ....);
```

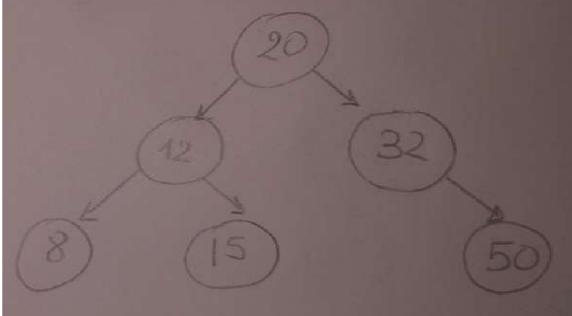
Scegli un'alternativa:

 a.

```
insert( 8, ...);
insert( 12, ...);
insert( 15, ...);
insert( 20, ...);
insert( 32, ...);
insert( 50, ...);
```

 b.

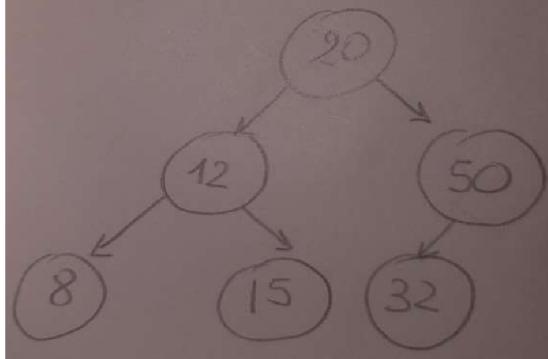
```
insert( 8, ...);
insert( 12, ...);
insert( 15, ...);
insert( 20, ...);
insert( 32, ...);
insert( 50, ...);
```



c.

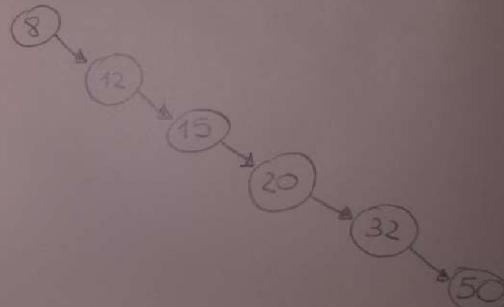
```
insert(8, ...);
insert(12, ...);
insert(15, ...);
insert(20, ...);
insert(32, ...);
insert(50, ...);
```

X



Risposta errata.

```
insert(8, ...);
insert(12, ...);
insert(15, ...);
insert(20, ...);
insert(32, ...);
insert(50, ...);
```



La risposta corretta è:

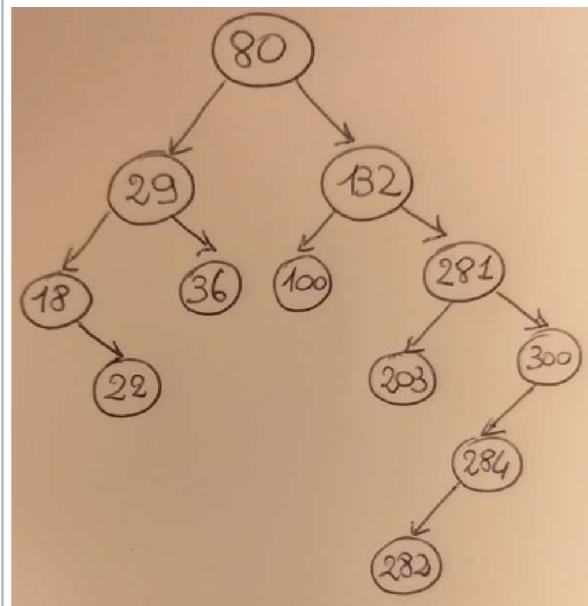
**Domanda 2**

Risposta corretta

Punteggio ottenuto 1,00 su 1,00

Si consideri il BST **b** disegnato sotto. Qual è il risultato delle seguenti chiamate a insert su **b** (indichiamo solo la chiave e trascuriamo il valore, rappresentato da puntini), eseguite nell'ordine dall'alto verso il basso (la prima chiamata è insert( 40, ...); ) in cui sono mostrate?

```
insert( 40, ...);
insert( 39, ...);
insert( 24, ...);
```

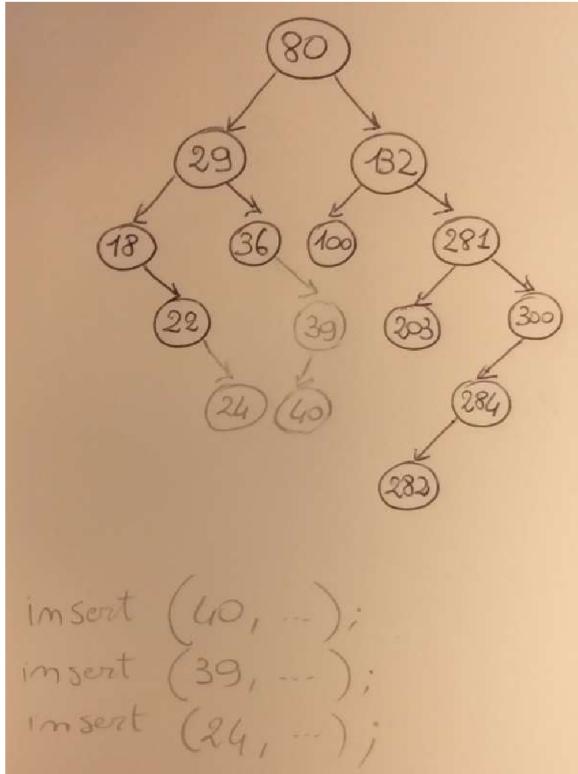


Scegli un'alternativa:

 a.

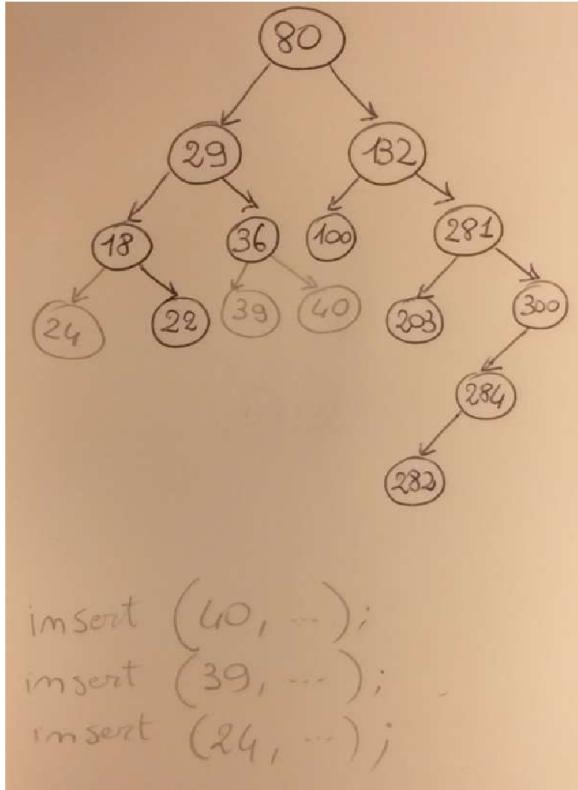
```
insert( 40, ...);
insert( 39, ...);
insert( 24, ...);
```

b.



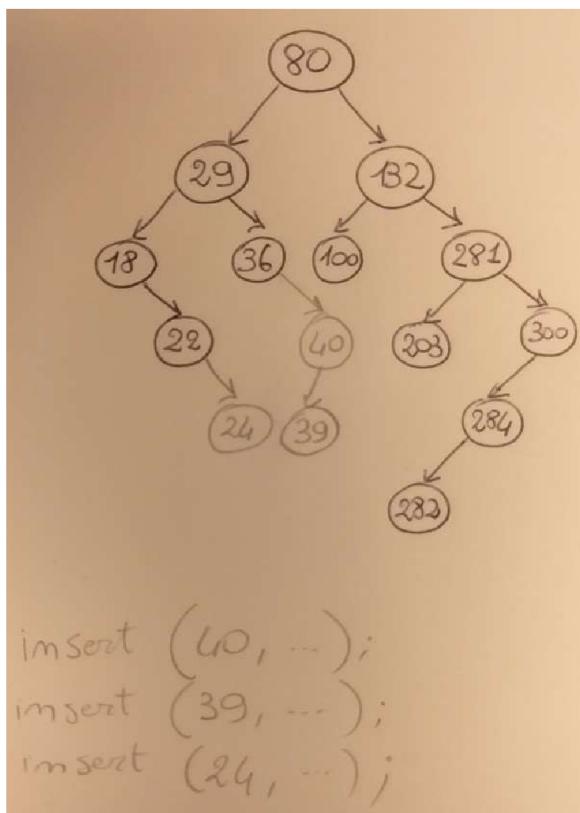
insert (40, ...);  
 insert (39, ...);  
 insert (24, ...);

c.



insert (40, ...);  
 insert (39, ...);  
 insert (24, ...);

Risposta corretta.



La risposta corretta è:

### Domanda 3

Risposta corretta

Punteggio ottenuto 1,00 su 1,00

Qual è la complessità di insert nel caso migliore?

Scegli un'alternativa:

- a. Il caso migliore è quello in cui il BST è un albero binario completo con altezza  $h = \log_2 n$ , dove  $n$  è il numero dei nodi. La complessità in questo caso è in Theta ( $\log n$ ).
- b. Nel caso migliore devo inserire l'elemento subito sotto la radice e questo ha costo Theta( 1 ) ; non importa quale sia il numero  $n$  di elementi presenti nel BST che implementa il dizionario: potrebbero esserci migliaia di elementi, ma ad esempio tutti nel sottoalbero a sinistra della radice: se devo inserire un elemento a destra della radice, questo inserimento ha costo costante perché "trovo subito il posto giusto in cui inserire".
- c. Il caso migliore è quello in cui il BST è vuoto e devo quindi inserire solo la radice, con complessità costante in Theta ( 1 ).

Risposta corretta.

La risposta corretta è: Nel caso migliore devo inserire l'elemento subito sotto la radice e questo ha costo Theta( 1 ) ; non importa quale sia il numero  $n$  di elementi presenti nel BST che implementa il dizionario: potrebbero esserci migliaia di elementi, ma ad esempio tutti nel sottoalbero a sinistra della radice: se devo inserire un elemento a destra della radice, questo inserimento ha costo costante perché "trovo subito il posto giusto in cui inserire".

**Domanda 4****Risposta errata**

Punteggio ottenuto -1,00 su 1,00

Qual è la complessità della search in un dizionario implementato come BST con altezza  $h$  e numero di nodi  $n$ , nel caso peggiore?

Scegli un'alternativa:

- a. Theta (  $n$  )
- b. Theta (  $\log n$  ) ✗
- c. Theta (  $h$  )

**Risposta errata.**

La risposta corretta è: Theta (  $h$  )

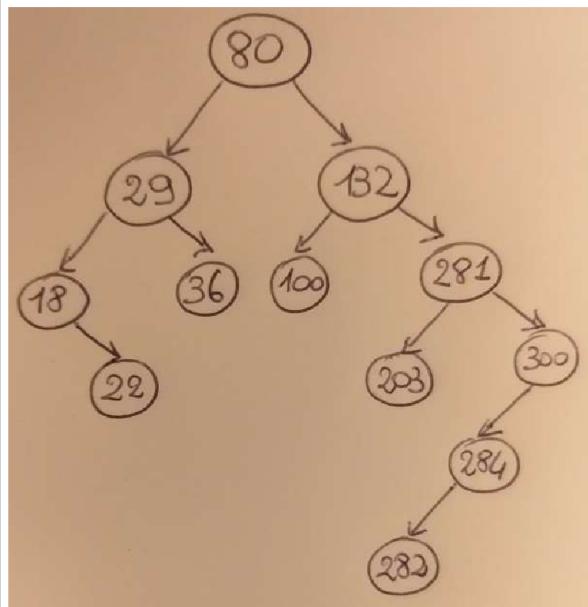
**Domanda 5**

Risposta errata

Punteggio ottenuto -1,00 su 1,00

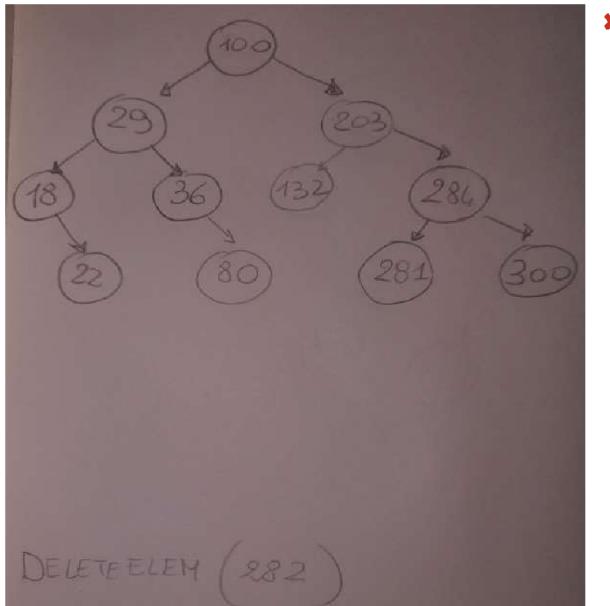
Si consideri il BST **b** disegnato sotto. Qual è il risultato della seguente chiamata a `deleteElem` su **b** (indichiamo solo la chiave e trascuriamo il valore, rappresentato da puntini)?

```
deleteElem( 282, ....);
```

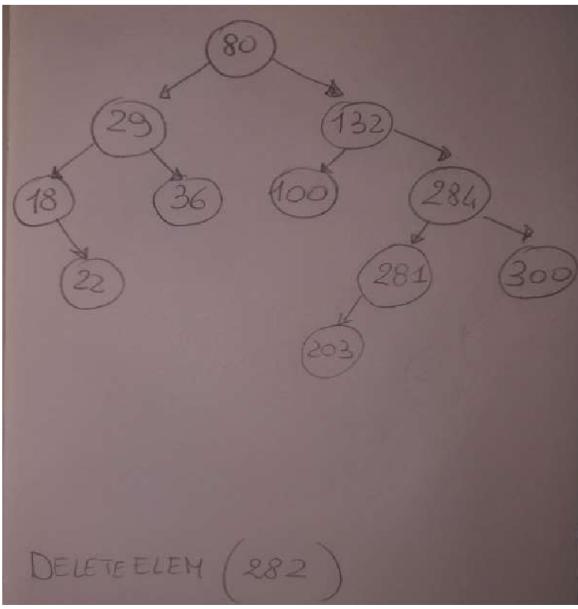


Scegli un'alternativa:

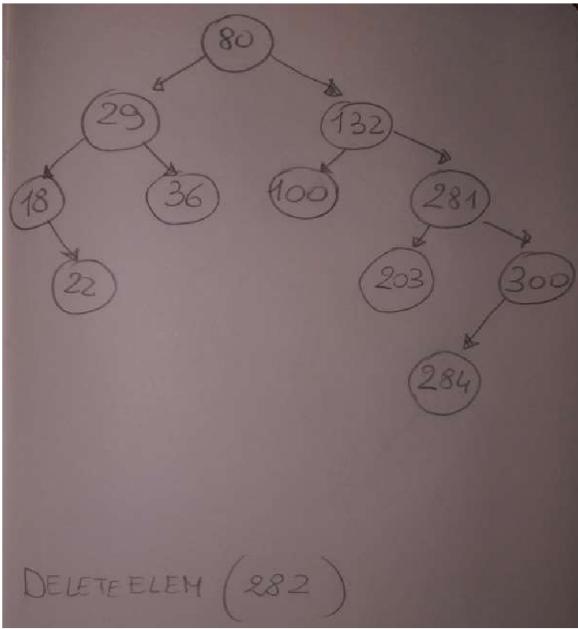
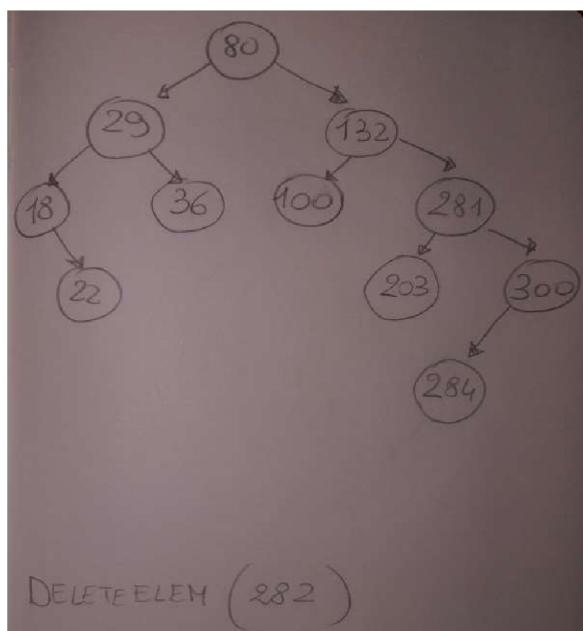
a.



○ b.



○ c.

**Risposta errata.****La risposta corretta è:**

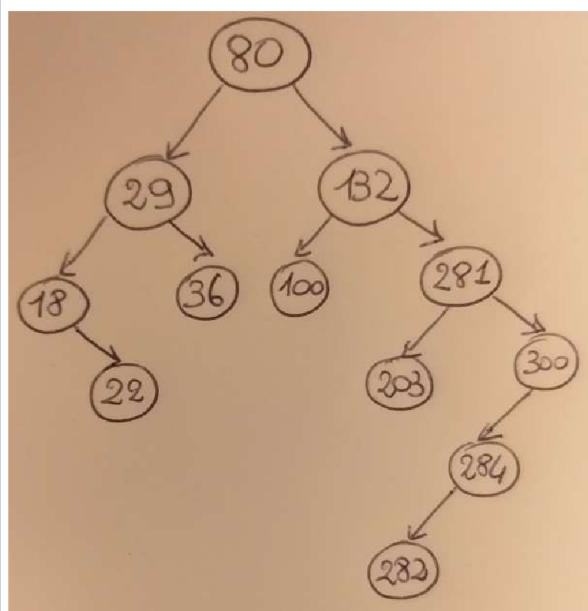
**Domanda 6**

Risposta errata

Punteggio ottenuto -1,00 su 1,00

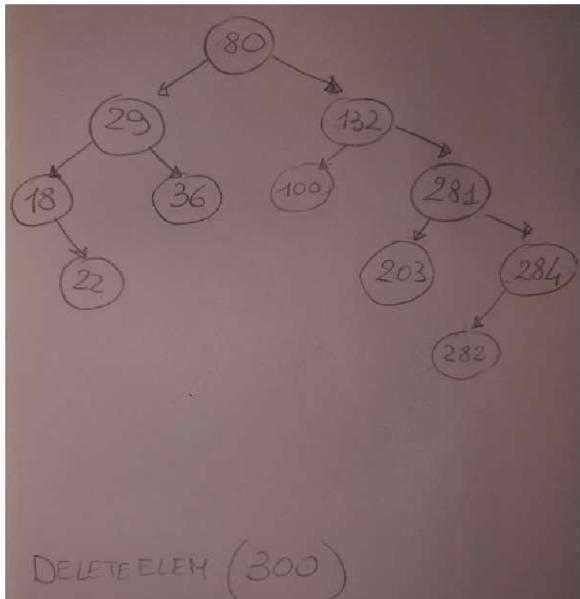
Si consideri il BST **b** disegnato sotto. Qual è il risultato della seguente chiamata a `deleteElem` su **b** (indichiamo solo la chiave e trascuriamo il valore, rappresentato da puntini)?

```
deleteElem( 300, ....);
```

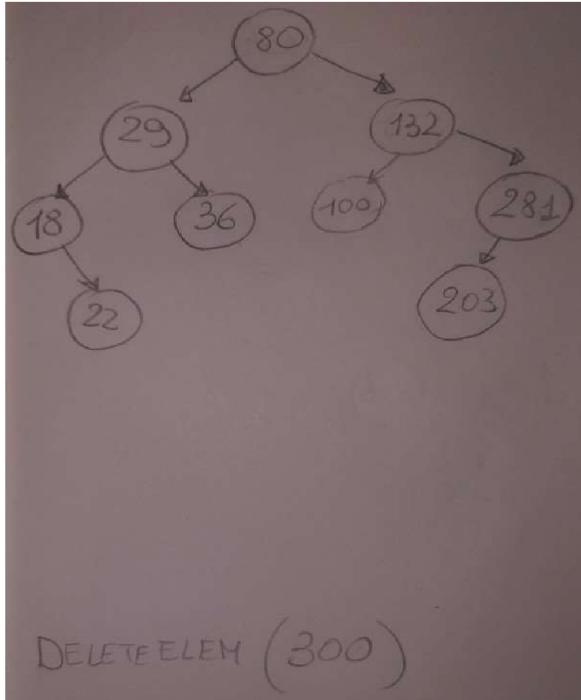


Scegli un'alternativa:

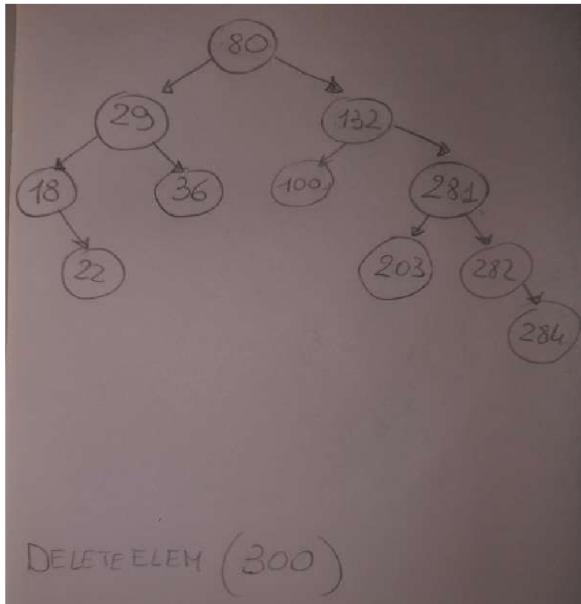
a.



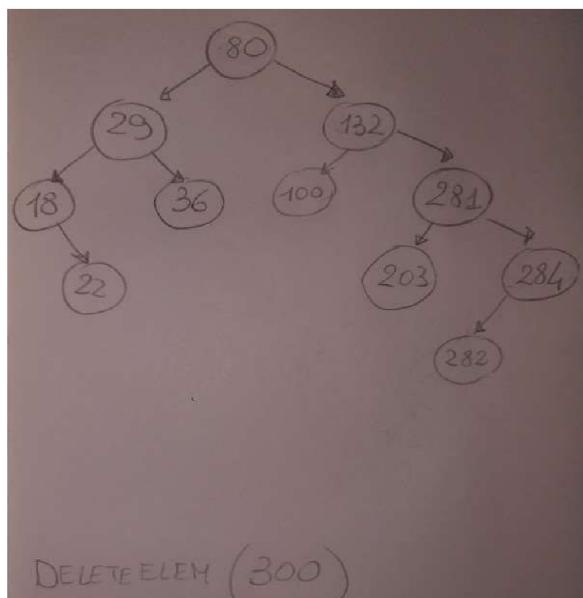
b.



c.



Risposta errata.



La risposta corretta è:

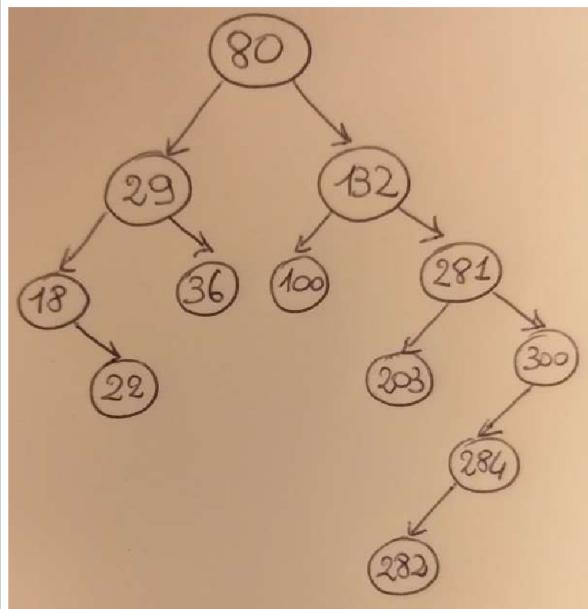
**Domanda 7**

Risposta errata

Punteggio ottenuto -1,00 su 1,00

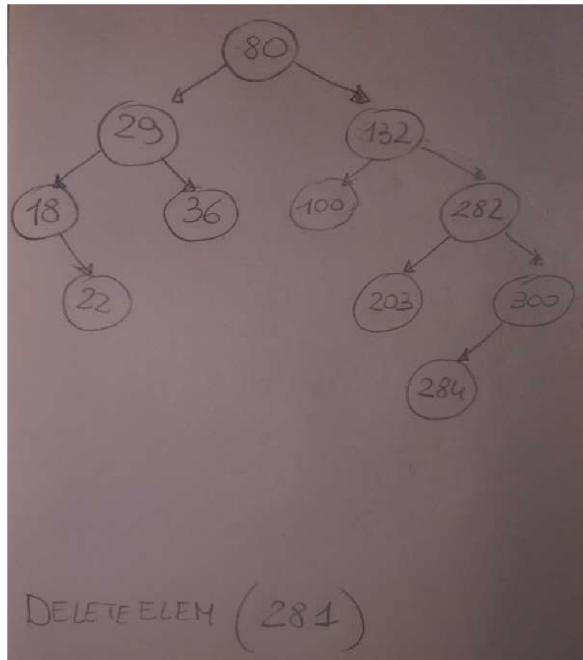
Si consideri il BST **b** disegnato sotto. Qual è il risultato della seguente chiamata a `deleteElem` su **b** (indichiamo solo la chiave e trascuriamo il valore, rappresentato da puntini)?

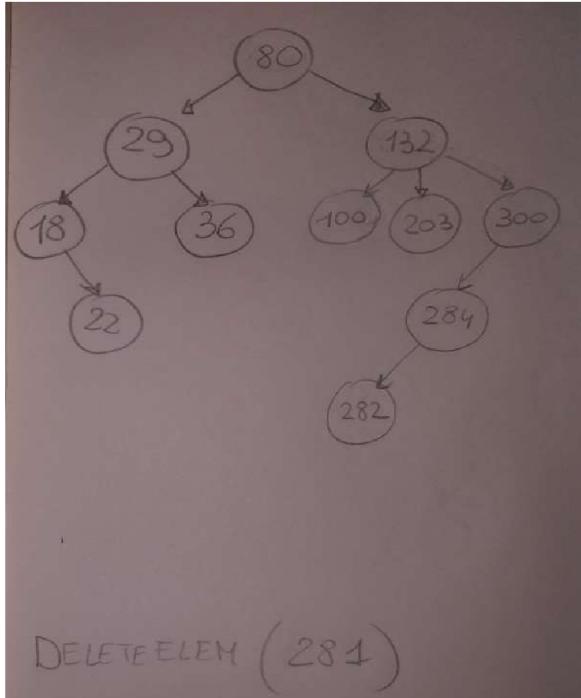
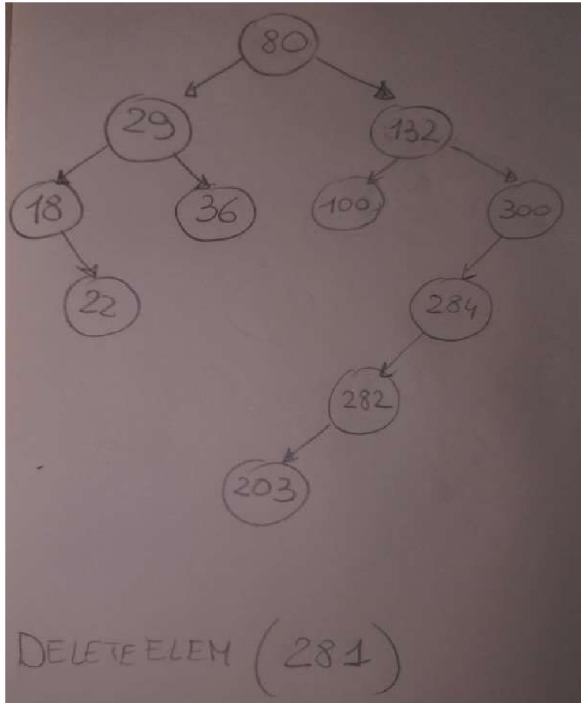
```
deleteElem( 281, ....);
```



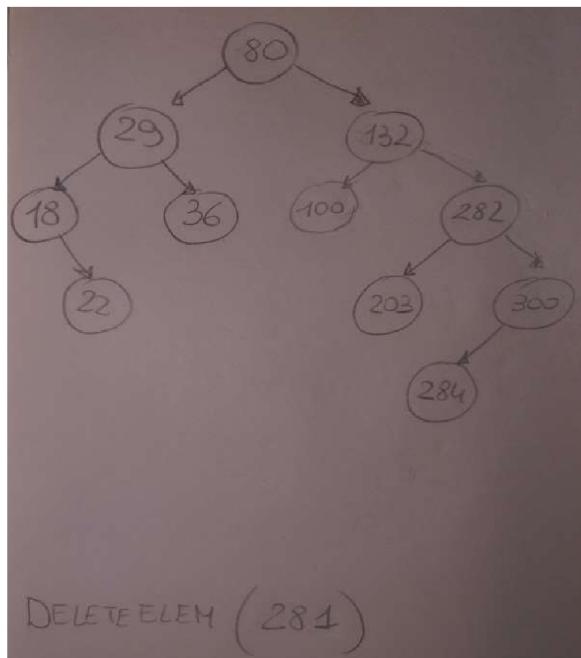
Scegli un'alternativa:

a.



b. c.

Risposta errata.



La risposta corretta è:

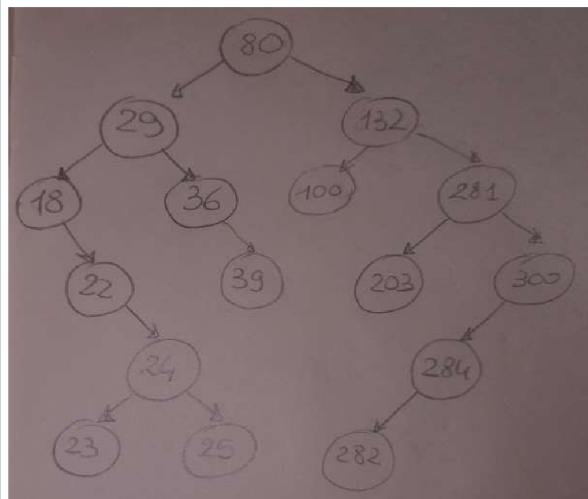
**Domanda 8**

Risposta errata

Punteggio ottenuto -1,00 su 1,00

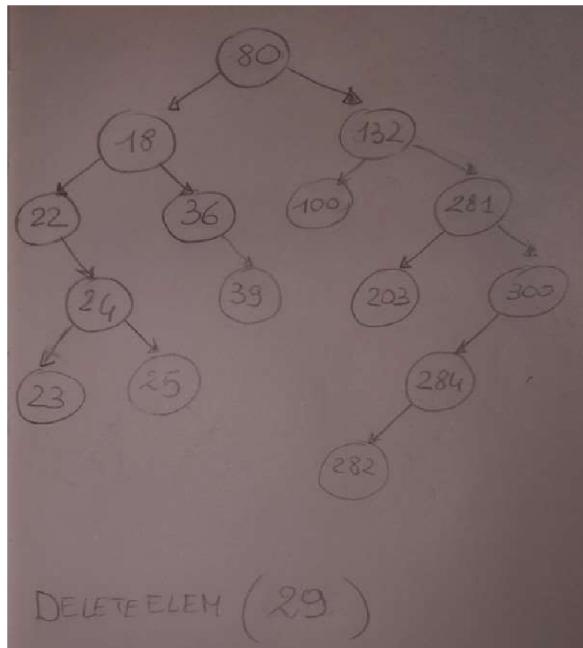
Si consideri il BST **b** disegnato sotto. Qual è il risultato della seguente chiamata a `deleteElem` su **b** (indichiamo solo la chiave e trascuriamo il valore, rappresentato da puntini)?

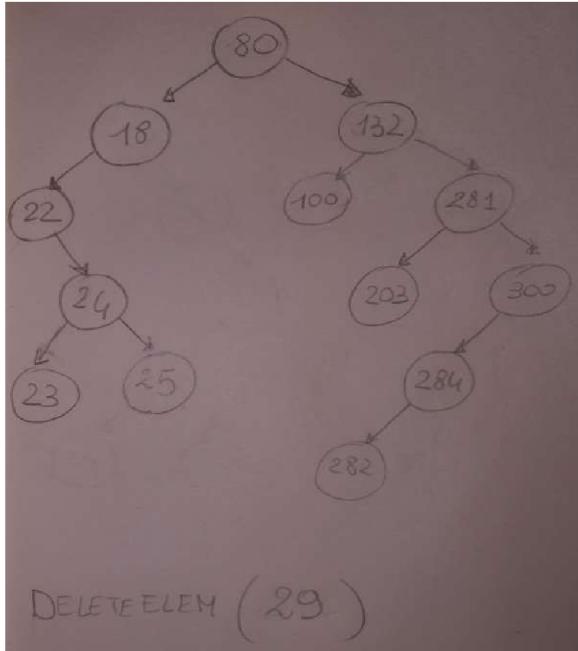
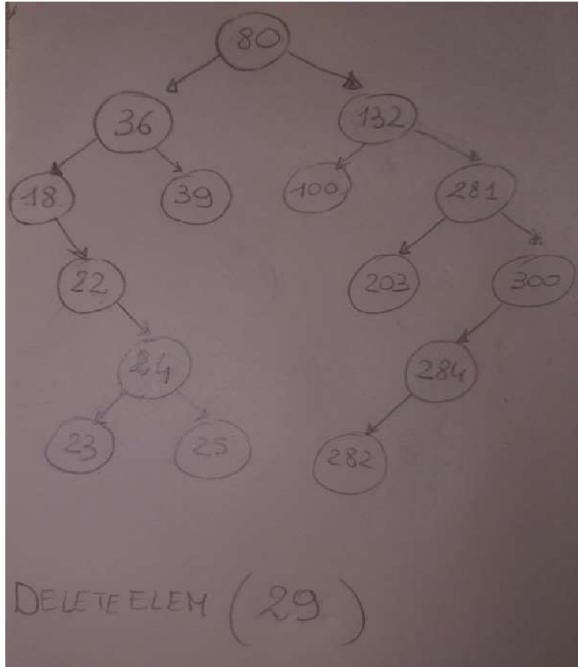
`deleteElem( 29, ....);`



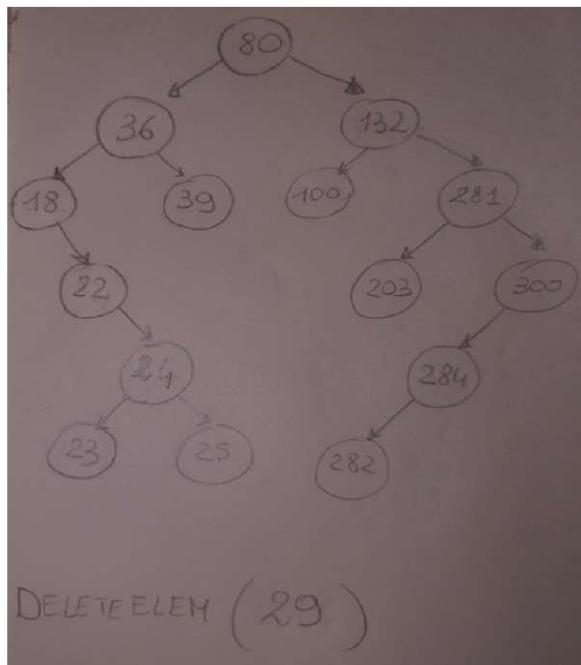
Scegli un'alternativa:

a.



b. c.

Risposta errata.



La risposta corretta è:

**Iniziato** mercoledì, 31 maggio 2023, 13:43

**Stato** Completato

**Terminato** mercoledì, 31 maggio 2023, 13:44

**Tempo impiegato** 18 secondi

**Punteggio** 0,00/2,00

**Valutazione** 0,00 su un massimo di 10,00 (0%)

**Domanda 1**

Risposta corretta

Punteggio ottenuto 1,00 su 1,00

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

Scegli un'alternativa:

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

Risposta corretta.

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

**Domanda 2**

Risposta errata

Punteggio ottenuto -1,00 su 1,00

Si supponga di dover stampare le etichette dei nodi di un albero implementato con struttura dati primo figlio-prossimo fratello; si supponga che `printLabel` sia la funzione che stampa l'etichetta di un nodo. Si consideri la seguente implementazione della funzione `f`:

```
void f(const Tree& t)
{
if (isEmpty(t)) return;
printLabel(t->label);
Tree auxT = t->firstChild;
while (!isEmpty(auxT))
{
    f(auxT);
    auxT=auxT->nextSibling;
}
}
```

Scegli un'alternativa:

- a. `f` non implementa correttamente una funzione finalizzata a stampare le etichette dei nodi perché non restituisce il valore da stampare
- b. `f` non implementa correttamente una funzione finalizzata a stampare le etichette dei nodi perché viene richiamata solo sul primo figlio di un nodo e non sugli altri.
- c. `f` implementa correttamente la funzione finalizzata a stampare le etichette dei nodi mediante una visita DFS in pre-ordine ✘
- d. `f` implementa correttamente la funzione finalizzata a stampare le etichette dei nodi mediante una visita DFS in post-ordine

Risposta errata.

`f` non implementa una visita esaustiva dell'albero perché non passa mai al "nextSibling": nel while, l'istruzione `auxT=auxT->nextSibling;` è sbagliata, dovrebbe essere `auxT=auxT->nextSibling;`

La risposta corretta è: `f` non implementa correttamente una funzione finalizzata a stampare le etichette dei nodi perché viene richiamata solo sul primo figlio di un nodo e non sugli altri.

**Iniziato** mercoledì, 31 maggio 2023, 13:44

**Stato** Completato

**Terminato** mercoledì, 31 maggio 2023, 13:44

**Tempo impiegato** 16 secondi

**Punteggio** -4,00/4,00

**Valutazione** -10,00 su un massimo di 10,00 (-100%)

### Domanda 1

Risposta errata

Punteggio ottenuto -1,00 su 1,00

Rispetto alla definizione delle operazioni del TDD Tree, l'approccio del libro di testo prevede di definirle direttamente sull' "insieme dei nodi".

Un altro approccio possibile è invece assumere che i nodi dell'albero siano etichettati con etichette in un insieme Label e che ogni nodo abbia etichetta unica e definire quindi le operazioni sul TDD Tree in termini delle etichette, viste come identificatori univoci dei nodi.

Quale delle seguenti considerazioni è più corretta?

Scegli un'alternativa:

- a. Entrambi gli approcci hanno pro e contro; un ragionevole compromesso è definire le operazioni del TDD Tree in termini di "etichette dei nodi" e non di "nodi" per renderle più astratte possibile, ma impiegare funzioni ausiliarie che operino direttamente sui nodi, in modo efficiente
- b. E' totalmente sbagliato usare nel TDD l'insieme dei nodi: il concetto di "nodo" è molto legato a quale struttura dati viene usata per implementare il TDD Tree (potrà essere un indice in un array, un puntatore a una struct, ...) quindi nella pratica si perde un livello di astrazione
- c. E' totalmente sbagliato usare nel TDD l'insieme delle etichette: definire le operazioni sul TDD Tree direttamente in termini dei nodi dell'albero permette implementazioni delle operazioni più efficienti, rispetto all'uso delle etichette che introducono un livello di astrazione intermedio

Risposta errata.

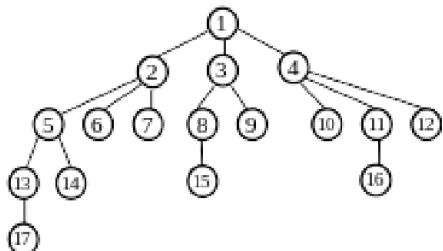
La risposta corretta è: Entrambi gli approcci hanno pro e contro; un ragionevole compromesso è definire le operazioni del TDD Tree in termini di "etichette dei nodi" e non di "nodi" per renderle più astratte possibile, ma impiegare funzioni ausiliarie che operino direttamente sui nodi, in modo efficiente

**Domanda 2**

Risposta errata

Punteggio ottenuto -1,00 su 1,00

Si consideri l'albero generico radicato in 1 disegnato sotto. In quale ordine vengono visitati i nodi con una DFS in pre-ordine?



Scegli un'alternativa:

- a. 17-16-14-13-11-10-8-7-5-4-2-1
- b. 1-2-3-4-5-6-7-8-9-10-11-12-13-14-15-16-17
- c. 1-2-5-13-17-14-6-7-3-8-15-9-4-10-11-16-12
- d. 1-2-5-13-17-5-14-2-6-2-7-1-3-8-15-3-9-1
- e. 1-2-6-7-5-3-9-8-7-3-8-15-9-4-10-11-16-12-17
- f. 17-16-15-14-13-12-11-10-9-8-7-6-5-4-3-2-1 ✗
- g. 17-13-14-5-6-7-2-15-8-9-3-10-16-11-12-4-1
- h. 17-13-5-2-1-2-6-7-1-8-7-6-5-4-3-2-1-10-11-4

Risposta errata.

La risposta corretta è:

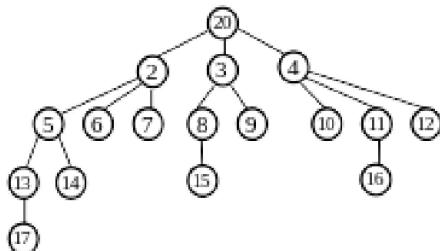
1-2-5-13-17-14-6-7-3-8-15-9-4-10-11-16-12

**Domanda 3**

Risposta errata

Punteggio ottenuto -1,00 su 1,00

Si consideri l'albero generico radicato in 20 disegnato sotto. In quale ordine vengono visitati i nodi con una DFS in **post-ordine** (ovvero, la radice di ogni sottoalbero viene visitata dopo la chiamata ricorsiva della visita su tutti i suoi figli)?



Scegli un'alternativa:

- a. 20-2-3-4-5-6-7-8-9-10-11-12-13-14-15-16-17
- b. 17-16-15-14-13-12-11-10-9-8-7-6-5-4-3-2-1 ✗
- c. 20-2-13-17-5-14-2-6-2-7-3-8-15-9-10-11-16-12
- d. 17-16-15-14-13-12-11-10-9-8-7-6-5-4-3-2-20
- e. 17-13-5-2-1-2-6-7-1-8-7-6-5-4-3-2-1-10-11-4
- f. 20-2-5-7-6-3-9-8-7-3-8-15-9-4-10-11-16-12-17
- g. 17-13-14-5-6-7-2-15-8-9-3-10-16-11-12-4-20
- h. 20-2-5-13-17-14-6-7-3-8-15-9-4-10-11-16-12

Risposta errata.

La risposta corretta è:

17-13-14-5-6-7-2-15-8-9-3-10-16-11-12-4-20

**Domanda 4**

Risposta errata

Punteggio ottenuto -1,00 su 1,00

Quale delle seguenti visite, visita i nodi di un BST in ordine crescente di etichetta?

Scegli un'alternativa:

- a. DFS simmetrica
- b. BFS
- c. DFS in post-ordine ✗
- d. DFS in pre-ordine

Risposta errata.

La risposta corretta è: DFS simmetrica

**Iniziato** mercoledì, 31 maggio 2023, 13:45

**Stato** Completato

**Terminato** mercoledì, 31 maggio 2023, 13:45

**Tempo impiegato** 33 secondi

**Valutazione** -8,00 su un massimo di 10,00 (-80%)

**Domanda 1**

Risposta errata

Punteggio ottenuto -1,00 su 1,00

Sia  $G=(V,E)$  un grafo orientato e sia  $(x,y)$  un elemento di  $E$ .

Scegli un'alternativa:

- a.  $y$  è adiacente a  $x$  ma  $x$  non è adiacente a  $y$
- b.  $y$  è adiacente a  $x$  e  $x$  è adiacente a  $y$  ✗
- c.  $(x,y)$  esce dal vertice  $y$  ed entra nel vertice  $x$

Risposta errata.

La risposta corretta è:  $y$  è adiacente a  $x$  ma  $x$  non è adiacente a  $y$

**Domanda 2**

Risposta errata

Punteggio ottenuto -1,00 su 1,00

In un grafo non orientato  $(V,E)$  con  $n$  vertici ed  $m$  archi, la somma del grado di tutti i vertici in  $V$  ( $\sum_{v \in V} \text{grado}(v)$ ) vale

Scegli un'alternativa:

- a.  $2m$
- b.  $2n$  ✗
- c.  $m$
- d.  $n$

Risposta errata.

La risposta corretta è:  $2m$

**Domanda 3**

Risposta errata

Punteggio ottenuto -1,00 su 1,00

In un grafo orientato  $(V, E)$  con  $n$  vertici ed  $m$  archi, la somma del grado uscente di tutti i vertici in  $V$  (ovvero  $\sum_{v \in V} \text{grado\_out}(v)$ ) vale

Scegli un'alternativa:

- a.  $m$
- b.  $2m$
- c.  $n$
- d.  $2n$  ✖

Risposta errata.

La risposta corretta è:  $m$

**Domanda 4**

Risposta errata

Punteggio ottenuto -1,00 su 1,00

Un cammino in un grafo è

Scegli un'alternativa:

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

Risposta errata.

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

**Domanda 5**

Risposta errata

Punteggio ottenuto -1,00 su 1,00

Un grafo non orientato  $G = (V, E)$  con  $n$  vertici ed  $m$  archi è connesso se e solo se

Scegli un'alternativa:

- a. per ogni coppia  $(v, w)$  di vertici di  $G$  esiste un cammino che li unisce
- b. esistono almeno  $2m$  coppie distinte  $(v, w)$  di vertici di  $G$  unite da un arco ✖
- c. esiste almeno una coppia  $(v, w)$  di vertici di  $G$  unita da un cammino

Risposta errata.

La risposta corretta è: per ogni coppia  $(v, w)$  di vertici di  $G$  esiste un cammino che li unisce

**Domanda 6****Risposta errata**

Punteggio ottenuto -1,00 su 1,00

Un albero libero è

Scegli un'alternativa:

- a. un grafo orientato e connesso ✗
- b. un grafo non orientato, connesso e senza cicli
- c. un grafo non orientato e senza cicli

**Risposta errata.**

La risposta corretta è: un grafo non orientato, connesso e senza cicli

**Domanda 7****Risposta errata**

Punteggio ottenuto -1,00 su 1,00

Se rimuoviamo un arco da un albero libero, il grafo risultante è

Scegli un'alternativa:

- a. non connesso
- b. non completo
- c. non ciclico ✗

**Risposta errata.**

La risposta corretta è: non connesso

**Domanda 8****Risposta errata**

Punteggio ottenuto -1,00 su 1,00

Se aggiungiamo un arco a un albero libero, il grafo risultante è

Scegli un'alternativa:

- a. completo ✗
- b. non connesso
- c. ciclico

**Risposta errata.**

La risposta corretta è: ciclico

**Domanda 9**

Risposta corretta

Punteggio ottenuto 1,00 su 1,00

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

Scegli un'alternativa:

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

Risposta corretta.

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

**Domanda 10**

Risposta errata

Punteggio ottenuto -1,00 su 1,00

La funzione addVertex del TDD Graph ha la seguente segnatura

Scegli un'alternativa:

- a. addVertex: Label x Weight x Graph -> Graph ✗  
(oltre al grafo da modificare, bisogna passare l'etichetta del nuovo nodo e il suo peso)
- b. addVertex: Label x Label x Graph -> Graph  
(oltre al grafo da modificare, bisogna passare l'etichetta del nodo padre a cui attaccare il nuovo nodo e l'etichetta del nuovo nodo)
- c. addVertex: Label x Graph -> Graph  
(oltre al grafo da modificare, bisogna passare solo l'etichetta del nuovo nodo)

Risposta errata.

La risposta corretta è: addVertex: Label x Graph -> Graph  
(oltre al grafo da modificare, bisogna passare solo l'etichetta del nuovo nodo)

**Iniziato** mercoledì, 31 maggio 2023, 13:45**Stato** Completato**Terminato** mercoledì, 31 maggio 2023, 13:46**Tempo impiegato** 24 secondi**Punteggio** -6,00/8,00**Valutazione** -7,50 su un massimo di 10,00 (-75%)**Domanda 1**

Risposta errata

Punteggio ottenuto -1,00 su 1,00

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 di **addEdge** dell'arco **(x,y)** è, nel caso peggiore

Scegli un'alternativa:

- a. Theta ( minimo( grado ( x ), grado ( y ) ) ) ❌
- b. Theta ( n )
- c. Theta( 1 )

Risposta errata.

La risposta corretta è: Theta ( n )

**Domanda 2**

Risposta errata

Punteggio ottenuto -1,00 su 1,00

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 **areAdjacent** chiamata sui vertici **x** e **y** è

Scegli un'alternativa:

- a. Theta( 1 ) nel caso migliore e Theta ( grado(x) ) nel caso peggiore
- b. Theta( 1 ) nel caso migliore e Theta ( grado(x) + grado(y) ) nel caso peggiore ❌
- c. Theta( 1 ) sia nel caso migliore che nel caso peggiore
- d. Theta ( grado(x) ) sia nel caso migliore che nel caso peggiore
- e. Theta( 1 ) nel caso migliore e Theta ( n ) nel caso peggiore
- f. Theta( 1 ) nel caso migliore e Theta ( min ( grado(x), grado(y) ) ) nel caso peggiore

Risposta errata.

La risposta corretta è: Theta( 1 ) nel caso migliore e Theta ( min ( grado(x), grado(y) ) ) nel caso peggiore

**Domanda 3**

Risposta corretta

Punteggio ottenuto 1,00 su 1,00

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 **addVertex** 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(1) nel caso migliore e Theta( $m$ ) nel caso peggiore

Risposta corretta.

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

**Domanda 4**

Risposta errata

Punteggio ottenuto -1,00 su 1,00

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 **degree** chiamata sul vertice  $x$  è

Scegli un'alternativa:

- a. Theta(1) sia nel caso migliore che nel caso peggiore ✗
- b. Theta(1) nel caso migliore e Theta(grado( $x$ )) nel caso peggiore
- c. Theta(grado( $x$ )) sia nel caso migliore che nel caso peggiore
- d. Theta(1) nel caso migliore e Theta( $n$ ) nel caso peggiore

Risposta errata.

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

**Domanda 5**

Risposta errata

Punteggio ottenuto -1,00 su 1,00

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  $x$  è

Scegli un'alternativa:

- a. Theta(1) nel caso migliore e Theta( $n$ ) nel caso peggiore
- b. Theta(grado( $x$ )) sia nel caso migliore che nel caso peggiore
- c. Theta(1) sia nel caso migliore che nel caso peggiore ✗
- d. Theta(1) nel caso migliore e Theta(grado( $x$ )) nel caso peggiore

Risposta errata.

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

**Domanda 6**

Risposta errata

Punteggio ottenuto -1,00 su 1,00

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 **removeVertex** del vertice  $v$  è

Scegli un'alternativa:

- a.  $\Theta(\text{grado}(v))$  nel caso migliore e peggiore
- b.  $\Theta(m)$  nel caso migliore e  $\Theta(\text{grado}(v))$  nel caso peggiore X
- c.  $\Theta(m)$  nel caso migliore e peggiore
- d.  $\Theta(n)$  nel caso migliore e peggiore
- e.  $\Theta(\text{grado}(v))$  nel caso migliore e  $\Theta(m)$  nel caso peggiore

Risposta errata.

La risposta corretta è:  $\Theta(\text{grado}(v))$  nel caso migliore e  $\Theta(m)$  nel caso peggiore**Domanda 7**

Risposta errata

Punteggio ottenuto -1,00 su 1,00

Si consideri la struttura dati per memorizzare un grafo con  $n$  vertici ed  $m$  archi che fa uso di liste di adiacenza. Indipendentemente dal fatto che i vertici siano memorizzati in un array o in una lista, se ogni lista di adiacenza è implementata come una lista singolarmente collegata, la struttura dati richiede uno spazio pari a

Scegli un'alternativa:

- a.  $n+m$  "blocchi" per un grafo orientato ed  $n+2m$  "blocchi" per un grafo non orientato
- b.  $n+2m$  "blocchi" per un grafo orientato ed  $n+m$  "blocchi" per un grafo non orientato X
- c.  $m+2n$  "blocchi" per un grafo orientato ed  $m+2n$  "blocchi" per un grafo non orientato

Risposta errata.

La risposta corretta è:  $n+m$  "blocchi" per un grafo orientato ed  $n+2m$  "blocchi" per un grafo non orientato**Domanda 8**

Risposta errata

Punteggio ottenuto -1,00 su 1,00

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 **removeEdge** dell'arco  $(x,y)$  è

Scegli un'alternativa:

- a.  $\Theta(1)$  nel caso migliore e  $\Theta(\text{grado}(x) + \text{grado}(y))$  nel caso peggiore
- b.  $\Theta(1)$  sia nel caso migliore che nel caso peggiore X
- c.  $\Theta(1)$  nel caso migliore e  $\Theta(n)$  nel caso peggiore

Risposta errata.

La risposta corretta è:  $\Theta(1)$  nel caso migliore e  $\Theta(\text{grado}(x) + \text{grado}(y))$  nel caso peggiore



**Iniziato** mercoledì, 31 maggio 2023, 13:46

**Stato** Completato

**Terminato** mercoledì, 31 maggio 2023, 13:47

**Tempo impiegato** 27 secondi

**Valutazione** -2,00 su un massimo di 10,00 (-20%)

**Domanda 1**

Risposta errata

Punteggio ottenuto -1,00 su 1,00

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 di `removeEdge` dell'arco  $(x,y)$  è, nel caso peggiore

Scegli un'alternativa:

- a. Theta (  $n$  )
- b. Theta ( grado (  $x$  ) + grado (  $y$  ) ) ✖
- c. Theta(  $m$  )

Risposta errata.

La risposta corretta è: Theta (  $n$  )

**Domanda 2**

Risposta corretta

Punteggio ottenuto 1,00 su 1,00

Si consideri la struttura dati "matrice di adiacenza" per rappresentare un grafo non orientato con  $n$  nodi ed  $m$  archi. La complessità di `removeVertex` del vertice  $v$  è, nel caso peggiore

Scegli un'alternativa:

- a. Theta (  $n^2$  ) ✓
- b. Theta( 1 )
- c. Theta (  $m$  )
- d. Theta ( grado (  $v$  ) )
- e. Theta (  $n$  )

Risposta corretta.

La risposta corretta è: Theta (  $n^2$  )

**Domanda 3**

Risposta errata

Punteggio ottenuto -1,00 su 1,00

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 di **addVertex** del vertice  $v$  è, nel caso peggiore

Scegli un'alternativa:

- a. Theta (  $n$  )
- b. Theta ( minimo( grado (  $x$  ), grado (  $y$  ) ) )
- c. Theta( 1 ) ✗

Risposta errata.

La risposta corretta è: Theta (  $n$  )**Domanda 4**

Risposta corretta

Punteggio ottenuto 1,00 su 1,00

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

Scegli un'alternativa:

- a. Theta( 1 ) ✓
- b. Theta (  $n$  )
- c. Theta ( minimo( grado (  $x$  ), grado (  $y$  ) ) )
- d. Theta (  $m$  )

Risposta corretta.

La risposta corretta è: Theta( 1 )

**Domanda 5**

Risposta errata

Punteggio ottenuto -1,00 su 1,00

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 di **addEdge** dell'arco  $(x,y)$  è, nel caso peggiore

Scegli un'alternativa:

- a. Theta ( minimo( grado (  $x$  ), grado (  $y$  ) ) ) ✗
- b. Theta (  $n$  )
- c. Theta( 1 )

Risposta errata.

La risposta corretta è: Theta (  $n$  )

**Domanda 6**

Risposta corretta

Punteggio ottenuto 1,00 su 1,00

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 di **incidentEdges** chiamata sul vertice  $x$  è, nel caso peggiore

Scegli un'alternativa:

- a.  $\Theta(1)$
- b.  $\Theta(n)$  ✓
- c.  $\Theta(\deg(x))$

Risposta corretta.

La risposta corretta è:  $\Theta(n)$ **Domanda 7**

Risposta errata

Punteggio ottenuto -1,00 su 1,00

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 di **degree** chiamata sul vertice  $x$  è, nel caso peggiore

Scegli un'alternativa:

- a.  $\Theta(\deg(x))$  ✗
- b.  $\Theta(1)$
- c.  $\Theta(n)$

Risposta errata.

La risposta corretta è:  $\Theta(n)$ **Domanda 8**

Risposta corretta

Punteggio ottenuto 1,00 su 1,00

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 di **removeVertex** del vertice  $v$  è, nel caso peggiore

Scegli un'alternativa:

- a.  $\Theta(\deg(v))$
- b.  $\Theta(n)$
- c.  $\Theta(n + m)$  ✓
- d.  $\Theta(m)$

Risposta corretta.

La risposta corretta è:  $\Theta(n + m)$

**Domanda 9**[Risposta errata](#)

Punteggio ottenuto -1,00 su 1,00

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 di **areAdjacent** chiamata sui vertici  $x$  e  $y$  è, nel caso peggiore

Scegli un'alternativa:

- a. Theta (  $n$  )
- b. Theta ( grado( $x$ ) ) X
- c. Theta( 1 )

**Risposta errata.**

La risposta corretta è: Theta (  $n$  )

**Domanda 10**[Risposta errata](#)

Punteggio ottenuto -1,00 su 1,00

Si consideri la struttura dati "matrici di adiacenza" per rappresentare un grafo non orientato con  $n$  nodi ed  $m$  archi. La complessità di **degree** chiamata sul vertice  $x$  è, nel caso peggiore

Scegli un'alternativa:

- a. Theta (  $n$  )
- b. Theta( 1 )
- c. Theta ( grado( $x$ ) ) X

**Risposta errata.**

La risposta corretta è: Theta (  $n$  )

**Iniziato** mercoledì, 31 maggio 2023, 13:47

**Stato** Completato

**Terminato** mercoledì, 31 maggio 2023, 13:47

**Tempo impiegato** 22 secondi

**Punteggio** 2,00/8,00

**Valutazione** 2,50 su un massimo di 10,00 (25%)

#### Domanda 1

Risposta errata

Punteggio ottenuto -1,00 su 1,00

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

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

Scegli un'alternativa:

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

Risposta errata.

La risposta corretta è: Theta ( 1 )

#### Domanda 2

Risposta corretta

Punteggio ottenuto 1,00 su 1,00

Si consideri il **TDD Coda con Priorità** implementato con una **lista semplice ordinata**.

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

Scegli un'alternativa:

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

Risposta corretta.

La risposta corretta è: Theta ( 1 )

**Domanda 3**

Risposta corretta

Punteggio ottenuto 1,00 su 1,00

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

Qual è la complessità dell'operazione **insertElem: Elem x Key x PriorityQueue -> PriorityQueue** nel caso peggiore?

Scegli un'alternativa:

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

Risposta corretta.

La risposta corretta è: Theta ( log n )

**Domanda 4**

Risposta corretta

Punteggio ottenuto 1,00 su 1,00

Uno heap binario è...

Scegli un'alternativa:

- a. un BST quasi completo
- b. un albero binario che soddisfa le seguenti proprietà
  - ogni nodo v contiene un elemento elem(v) cui è associata una chiave chiave(v) presa da un dominio totalmente ordinato
  - le chiavi degli elementi nel sottoalbero sinistro di v sono  $\leq$  chiave(v)
  - le chiavi degli elementi nel sottoalbero destro di v sono  $\geq$  chiave(v)
- c. un albero binario quasi completo tale che
  - ogni nodo v contiene un elemento elem(v) ed una chiave chiave(v) presa da un dominio totalmente ordinato
  - il valore della chiave in un nodo v è maggiore dei valori delle chiavi associate ai suoi due figli✓

Risposta corretta.

La risposta corretta è: un albero binario quasi completo tale che

- ogni nodo v contiene un elemento elem(v) ed una chiave chiave(v) presa da un dominio totalmente ordinato
- il valore della chiave in un nodo v è maggiore dei valori delle chiavi associate ai suoi due figli

**Domanda 5**

Risposta errata

Punteggio ottenuto -1,00 su 1,00

Si consideri il **TDD Coda con Priorità** implementato con una **lista semplice ordinata**.

Qual è la complessità dell'operazione **insertElem: Elem x Key x PriorityQueue -> PriorityQueue** nel caso peggiore?

Scegli un'alternativa:

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

Risposta errata.

La risposta corretta è: Theta ( n )

**Domanda 6**

Risposta corretta

Punteggio ottenuto 1,00 su 1,00

Si consideri il **TDD Coda con Priorità** implementato con una **lista semplice ordinata**.

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

Scegli un'alternativa:

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

Risposta corretta.

La risposta corretta è: Theta ( 1 )

**Domanda 7**

Risposta corretta

Punteggio ottenuto 1,00 su 1,00

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

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

Scegli un'alternativa:

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

Risposta corretta.

La risposta corretta è: Theta ( log n )

**Domanda 8**

Risposta errata

Punteggio ottenuto -1,00 su 1,00

**I BST e gli heap binari**

Scegli un'alternativa:

- a. sono entrambi alberi generici in cui l'altezza è sempre al più  $\log_2 n$
- b. sono entrambi alberi binari quasi completi
- c. sono entrambi alberi binari completi
- d. sono entrambi alberi generici in cui l'altezza può variare da  $n$  (con  $n$  numero dei nodi) a  $n^2$  ✗
- e. sono entrambi alberi binari

Risposta errata.

La risposta corretta è: sono entrambi alberi binari

