Pode previsualizar este teste mas se fosse uma tentativa real seria bloqueado porque:

Este teste só está acessível a partir de certas redes e este computador não se encontra numa delas.

Este teste não está disponível

Pergunta 1

Por responder Pontuação 1,000

Imagine que está a **ordenar de forma decrescente** um *vector* de inteiros usando a seguinte implementação do *BubbleSort*:
/
*Imagine you are **sorting in decreasing order** an integer* vector *using the following implementation of* BubbleSort*:*

```
void bubblesort(vector<int> & v) {
  for (unsigned i=0; i<v.size()-1; i++)
    for (unsigned j=0; j<v.size()-i-1; j++)
      if (v[j] < v[j+1])
        swap(v[j], v[j+1]);
}
```

Qual das seguintes condições seria um **invariante útil no início de cada iteração** para provar a correção do ciclo com a variável i ?
/
*Which of the following conditions would be an **useful invariant at the start of each iteration** to prove the correction of the cycle with the variable i ?*

- a. Os **mais pequenos** i números de todo o *vector* estão nas **primeiras** i posições e estão ordenados
  /
  *The **smallest** i numbers of the entire* vector *are on the **first** i positions and are sorted*
- b. Os **maiores** i números de todo o *vector* estão nas **primeiras** i posições e estão ordenados
  /
  *The **biggest** i numbers of the entire* vector *are on the **first** i positions and are sorted*
- c. Os **mais pequenos** i números de todo o *vector* estão nas **últimas** i posições e estão ordenados
  /
  *The **smallest** i numbers of the entire* vector *are on the **last** i positions and are sorted*
- d. Os **maiores** i números de todo o *vector* estão nas **últimas** i posições e estão ordenados
  /
  *The **biggest** i numbers of the entire* vector *are on the **last** i positions and are sorted*

Pergunta 2

Por responder Pontuação 1,000

Considere as seguintes 2 funções:
/
Consider the following 2 functions:

- $f(n) = 2^n$
- $g(n) = n^2$

A notação (g(n)) representa o conjunto de funções que crescem assintoticamente de forma igual ou proporcional a g(n)

Como f(n) é proporcional a n, mas não ao quadrado de n, a relação correta é f(n)(g(n))

Qual das afirmações é **verdadeira**?
/
*Which of the statements is **true**?*

- a. $f(n) \in \Theta(g(n))$
- b. $f(n)$ não é Θ, $\mathcal{O}$ ou Ω de $g(n)$
  \
  $f(n)$ is not Θ, $\mathcal{O}$ or Ω of $g(n)$
- c. $f(n) \in \mathcal{O}(g(n))$
- ⊗ d. $f(n) \in \Omega(g(n))$

Pergunta 3

Por responder Pontuação 1,000

Considere a seguinte **função recursiva** em C++ para calcular a soma dos números inteiros entre *a* e *b*:
/
*Consider the following C++ **recursive function** to compute the sum of the integer numbers between* a *and* b*:*

```
int sum(int a, int b) {
  if (a==b) return a;
  int middle = (b-a)/2 + a;
  return sum(a, middle) + sum(middle+1, b);

}
```

Qual é a sua **complexidade temporal**?
/
*What is its **temporal complexity**?*

- a. $\Theta(n^2)$
- ⊗ b. $\Theta(n \log n)$
- c. $\Theta(\log n)$
- d. $\Theta(n)$

Resposta d)

Pergunta 4

Por responder Pontuação 1,000

Suponho que tem um programa de **complexidade temporal** $\mathcal{O}(n^3)$ que demora **2 segundos** para um input de tamanho $n = 1000$
\
*Suppose you have a program with **temporal complexity** $\mathcal{O}(n^3)$ that takes **2 seconds** for an input of size $n = 1000$*

Quanto segundos **estima** que demoraria num input de tamanho $n = 2000$?
\
How many seconds do you **estimate** it would take on an input of size $n = 2000$?

- a. 8s
- b. 16s b)
- c. 32s
- d. 4s

Pergunta 5

Por responder Pontuação 1,000

Qual a vantagem da ordenação por contagem (C*ountingSort*) em relação à ordenação por partição (Q*uickSort*)?

/

What is the advantage of sorting by counting (CountingSort) over sorting by partition (QuickSort)?

- a. Ordenação por contagem (C*ountingSort*) tem menor complexidade espacial, sempre

  /

  CountingSort has less spatial complexity, always
- b. Ordenação por contagem (C*ountingSort*) tem menor complexidade temporal, apenas quando a gama de valores dos elementos a ordenar (*max-min*) é comparável ao número de elementos

  /

  CountingSort has less temporal complexity, only when the range of values of the elements to be sorted (max-min) is comparable to the number of elements
- c. Ordenação por contagem (C*ountingSort*) tem menor complexidade temporal, sempre

  /

  CountingSort has less temporal complexity, always
- d. Não existe vantagem, nunca

  /

  There is no advantage, ever

Pergunta 6

Por responder Pontuação 1,000

Uma lista ligada (duplamente ligada) mantém os seus elementos ordenados por ordem crescente. Usando o algoritmo mais adequado para pesquisa de um elemento nesta lista, a complexidade temporal da operação de pesquisa é:

/

A linked list (doubly linked) keeps its elements sorted in ascending order. Using the most appropriate algorithm for searching for an element in this list, the time complexity of the search operation is:

- a. $O(\log n)$
- b. $O(1)$
- c. $O(n)$
- d. $O(n \times \log n)$

Pergunta 7

Por responder Pontuação 1,000

Qualquer browser possui a funcionalidade de retorno à página visitada anteriormente. Na implementação desta (e apenas desta) funcionalidade, qual a estrutura de dados que considera mais adequada (eficiente em tempo e espaço)?

/

Any browser has the functionality to return to the previously visited page. When implementing this (and only this) functionality, which data structure do you consider most appropriate (efficient in time and space)?

- a. Fila

  /

  Queue
- b. Lista duplamente ligada

  /

  Doubly linked list
- c. Lista simplesmente ligada

  /

  Singly linked list
- d. Pilha

  /

  Stack

Pergunta 8

Por responder Pontuação 1,000

Se introduzir a sequência 10, 15, 7, 9, 17, 12 numa árvore binária de pesquisa, a árvore resultante:

/

If you add the sequence 10, 15, 7, 9, 17, 12 into a binary search tree, the resulting tree:

a. Não é completa, e é equilibrada

/

Is not complete, and is balanced

b. É completa, e não é equilibrada

/

Is complete and is not balanced

c. Não é completa e não é equilibrada

/

Is not complete and is not balanced

d. É completa e equilibrada

/

Is complete and balanced

Pergunta 9

Por responder Pontuação 1,000

Sobre iteradores implementados na STL para as estruturas lineares vetores, listas, pilhas e filas, é correto afirmar que:

/

Regarding iterators implemented in the STL for the linear structures vectors, lists, stacks and queues, it is correct to say that:

a. Iteradores são implementados em vetores e listas, apenas

/

Iterators are implemented on vectors and lists, only

b. Iteradores são usados em todas estas estruturas de dados, mas implementados de forma distinta

/

Iterators are used in all these data structures, but implemented differently

c. Iteradores são usados e implementados de forma similar em todas estas estruturas de dados

/

Iterators are used and implemented similarly in all these data structures

d. Iteradores são implementados em vetores, apenas

/

Iterators are implemented on vectors, only

Resposta A nas stacks e nas filas não temos iteradores .

Nas filas só temos acesso ao first-in e first-out

Na stack só temos acesso ao ultimo elemento, porque colocamos em baixo e tiramos por cima

Pergunta 10

Por responder Pontuação 1,000

Considere a estrutura **lista** implementada por uma lista simplesmente ligada com referências para o início e fim (primeiro e último nó da lista).

Quais das seguintes operações podem ser realizadas em tempo O(1)?

O(1) I. inserir um elemento no início da lista
O(1) II. inserir um elemento no fim da lista
O(1) III. remover o primeiro elemento (início) da lista
O(n) IV. remover o último elemento (fim) da lista

Tenho que percorrer o vetor até chegar ao penultimo elemento da lista

/

*Consider the **list** structure implemented by a simply linked list with references to the start and end (first and last node of the list).*

Which of the following operations can be performed in O(1) time?

I. insert an element at the beginning of the list

II. insert an element at the end of the list

III. remove the first element (start) of the list

IV. remove the last element (end) of the list

Resposta C

a. I, II e IV, apenas
/
I, II and IV, only

b. I e II, apenas
/
I and II, only

c. I, II, e III, apenas
/
I, II, and III, only

d. I e III, apenas
/
I and III, only