

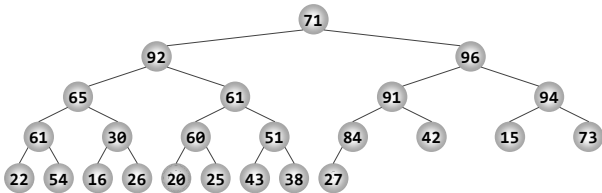


1. [0,5] Analise as afirmações a seguir e indique a alternativa correta.
 - I. *O Bubble Sort estende o sufixo ordenado de uma sequência, trocando itens quaisquer fora de ordem em seu prefixo.*
 - II. *O Selection Sort estende o prefixo ordenado de uma sequência, selecionando e trocando dois itens em seu sufixo.*
 - III. *O Insertion Sort estende o sufixo ordenado de uma sequência, inserindo nele um item removido de seu prefixo.*

(a) Apenas as afirmações I e II são verdadeiras.
 (b) Apenas as afirmações I e III são verdadeiras.
 (c) Apenas as afirmações II e III são verdadeiras.
 (d) Nenhuma das anteriores.
2. [0,5] Seja $\langle 0,1,3,5,7 \mid 9,2,6,4 \rangle$ uma sequência parcialmente ordenada, onde a barra separa o prefixo ordenado do sufixo desordenado. Quantas comparações entre itens o *Insertion Sort* precisa efetuar para estender o prefixo ordenado com mais um item?

(a) 1.
 (b) 2.
 (c) 3.
 (d) Nenhuma das anteriores.
3. [0,5] Analise as afirmações a seguir e indique a alternativa correta.
 - I. *A prévia ordenação da sequência que será ordenada afeta o desempenho do Bubble Sort e também do Insertion Sort.*
 - II. *Quando a sequência a ser ordenada tem muita repetição, o Insertion Sort pode ser menos eficiente que o Selection Sort.*
 - III. *No pior caso, o número de trocas feitas pelo Bubble Sort e pelo Insertion Sort é $O(n^2)$.*

(a) Apenas as afirmações I e II são verdadeiras.
 (b) Apenas as afirmações I e III são verdadeiras.
 (c) Apenas as afirmações II e III são verdadeiras.
 (d) Nenhuma das anteriores.
4. [0,5] Quantas trocas são necessárias para transformar o quase-monte de máximos a seguir em um monte de máximos?



- (a) 1.
 (b) 2.
 (c) 3.
 (d) Nenhuma das anteriores.

5. [0,5] Seja $\langle 16,15,14,13,12,10,9,11,8,7,6,4,5,2,3,1,0 \mid 17 \rangle$ uma sequência cujo prefixo é um monte de máximos e cujo sufixo é uma sequência ordenada. Quantas trocas o *Heap Sort* precisa efetuar para estender esse sufixo ordenado com mais um item, mantendo o prefixo como um monte de máximos?

(a) 2.
 (b) 3.
 (c) 4.
 (d) Nenhuma das anteriores.
6. [0,5] Analise as afirmações a seguir e indique a alternativa correta.
 - I. *O Quick Sort exige memória auxiliar proporcional à entrada.*
 - II. *A eficiência do Quick Sort varia entre $O(n \lg n)$ e $O(n^2)$.*
 - III. *A versão probabilística do Quick Sort reduz a probabilidade do algoritmo consumir tempo $O(n^2)$, no pior caso.*

(a) Apenas as afirmações I e II são verdadeiras.
 (b) Apenas as afirmações I e III são verdadeiras.
 (c) Apenas as afirmações II e III são verdadeiras.
 (d) Nenhuma das anteriores.
7. [0,5] Analise as afirmações a seguir e indique a alternativa correta.
 - I. *Self-organizing Linear Search é mais eficiente que Linear Search quando a distribuição de probabilidade de busca dos itens da sequência não é uniforme.*
 - II. *Interpolation Search é mais eficiente que Binary Search quando os itens da sequência em que a busca será feita estão distribuídos uniformemente.*
 - III. *Linear Search só funciona com sequências desordenadas e Binary Search só funciona com sequências ordenadas.*

(a) Apenas as afirmações I e II são verdadeiras.
 (b) Apenas as afirmações I e III são verdadeiras.
 (c) Apenas as afirmações II e III são verdadeiras.
 (d) Nenhuma das anteriores.
8. [0,5] Sendo $p(17) = 0.01$, $p(20) = 0.07$, $p(39) = 0.10$, $p(46) = 0.21$ e $p(58) = 0.61$ as probabilidades de busca dos itens da sequência $v = \langle 58,17,20,39,46 \rangle$, qual o número médio de comparações feitas pela busca linear para encontrar um item em v ?

(a) 1.66.
 (b) 2.29.
 (c) 3.43.
 (d) 4.34.

9. [2,0] Simule a execução da chamada `bb(34, [12,23,34,45,56,67,78,89, 90], 0, 8)`, preenchendo a tabela a seguir.

```
def bb(x, v, p, u)
  if p>u: return False
  m = (p+u)//2
  if x<v[m]: return bb(x, v, p, m-1)
  if x>v[m]: return bb(x, v, m+1, u)
  return True
```

x	p	u	m	v ₀	v ₁	v ₂	v ₃	v ₄	v ₅	v ₆	v ₇	v ₈

10. Implementação recursiva de um algoritmo de ordenação: *Bubble Sort*, *Selection Sort* ou *Insertion Sort*.

10.1. [2,0] Implementação recursiva da função auxiliar usada pelo algoritmo: `empurra(v, n)`, `seleciona(v, n)` ou `insere(x, v, u)`.

10.2. [2,0] Implementação recursiva da função principal do algoritmo: `bsort(v, n)`, `ssort(v, n)` ou `isort(v, n)`.

Obs.: A solução para essa questão do simulado não será divulgada; mas as dúvidas sobre ela poderão ser esclarecidas na aula.