**TESTE ALGORITMO:**

**Instruções para o teste:**

**Antes de começar o teste se apresente, fale a quanto tempo você está na Neon, o que você faz aqui e o que você vai fazer com o candidato. “*Ex: Meu nome é Mariana Nicolopulos, estou há 1,5 na Neon, faço parte do TeamX que é nosso time de RH. Vou aplicar agora o teste de algoritmo com você...*”**

* Escrever os algoritmos em Portugol ou fluxograma ou em uma das seguintes linguagens: Node.js, .Net, Go, Python.
* Será considerada a complexidade do algoritmo
* Tempo de teste: 45min a 60min
* Fazer uma pergunta (ou mais, quando necessário) de cada nível
* Pedir para o candidato enviar o link do teste resolvido a cada questão.

**1º NÍVEL:**

1. **Escreva um algoritmo que receba um texto como entrada e precisará retornar se ele é ou não um palíndromo. Sendo 1 para o caso de ser palíndromo e 0 quando não for.**

**Explicação:** O texto pode ser lido tanto da direita para a esquerda como da esquerda para a direita

1. **Escreva um algoritmo que exiba a sequência de Fibonnacci iniciando em 1 até o "n" elemento.**

**Explicação**: Na sequência de Fibonnacci, cada elemento é a soma dos dois elementos anteriores. Os dois primeiros elementos são 1 e 1.

Recebe como entrada o valor de “n”.

1. **Escreva um algoritmo que encontre o maior e o menor número de uma lista de inteiros.**

**Explicação**: Dado uma lista de inteiros como entrada, o algoritmo precisará retornar o maior e o menor número dentro da lista de inteiros.

**2º NÍVEL:**

1. **Escreva um algoritmo que some números pares.**

**Explicação**: Dada uma lista de números distintos encontre a soma dos números pares que resulte no menor valor, e deve retornar este valor.

1. **Escreva um algoritmo que simule o funcionamento de um caixa eletrônico.**

**Explicação**: Deverá receber o valor desejado de saque e ele retornará à quantidade de cédulas de cada valor. As cédulas consideradas pelo caixa eletrônico são: 100, 50, 20, 10, 5 e 2. Deve ser retornado a menor quantidade de cédulas possível para o valor do saque.

**3º NÍVEL:**

1. **Escreva um algoritmo de ordenação (Insertion Sort)**

**Explicação**: Dado uma lista de inteiros como entrada, o algoritmo precisará ordenar os valores utilizando o método de ordenação "Insertion Sort".

Insertion Sort, ou ordenação por inserção, é um algoritmo de ordenação que, dado uma estrutura (array, lista) constrói uma matriz final com um elemento de cada vez, uma inserção por vez. Assim como algoritmos de ordenação quadrática, é bastante eficiente para problemas com pequenas entradas, sendo o mais eficiente entre os algoritmos desta ordem de classificação.

Podemos fazer uma comparação do Insertion Sort com o modo como algumas pessoas organizam um baralho num jogo de cartas. Imagine que você está jogando às cartas. Você está com as cartas na mão e elas estão ordenadas. Você recebe uma nova carta e deve colocá-la na posição correta da sua mão de cartas, de forma a que as cartas obedeçam à ordenação.

A cada nova carta adicionada à sua mão de cartas, a nova carta pode ser menor que algumas das cartas que você já tem na mão ou maior, e assim, você começa a comparar a nova carta com todas as cartas na sua mão até encontrar sua posição correta. Você insere a nova carta na posição correta, e, novamente, a sua mão é composta de cartas totalmente ordenadas. Então, você recebe outra carta e repete o mesmo procedimento. Então outra carta, e outra, e assim em diante, até não receber mais cartas (fonte: Wikipedia).

- Desenhar junto com candidato. Explicar e fazer junto com ele ao longo da questão.

1. **Escreva um algoritmo que transforme a árvore a seguir em uma árvore binária. Explicação: Uma árvore binária é uma estrutura de dados caracterizada por:**
   1. Ou não tem elemento algum (árvore vazia).
   2. Ou tem um elemento distinto, denominado raiz, com dois ponteiros para duas estruturas diferentes, denominadas subárvore esquerda e subárvore direita.

Perceba que a definição é recursiva e, devido a isso, muitas operações sobre árvores binárias utilizam recursão. É o tipo de árvore mais utilizado na computação. A principal utilização de árvores binárias são as árvores binárias de busca. Os nós de uma árvore binária possuem graus zero, um ou dois. Um nó de grau zero é denominado folha. Em uma árvore binária, por definição, cada nó poderá ter até duas folhas, sendo que ela se compara com a ABB (árvore binária de busca), apesar de não ter a propriedade da mesma ("na abb, existe uma regra na inserção"). A profundidade de um nó é a distância deste nó até a raiz. Um conjunto de nós com a mesma profundidade é denominado nível da árvore. A maior profundidade de um nó, é a altura da árvore. Uma árvore "estritamente binária" é uma árvore na qual todo nó tem zero ou duas folhas. Existem autores, porém, que adotam essa definição para o termo quase completa, e utilizam o termo completa apenas para árvores em que todos os níveis têm o máximo número de elementos (fonte: Wikipedia).

