1-Estruturas de dados, tais como filas e pilhas, são utilizadas em diversas aplicações para automação industrial por meio de linguagens de programação textuais. O texto estruturado (ST) é uma das opções de linguagem de programação definidas pela norma IEC 61131-3. O trecho de código a seguir foi implementado nesse contexto.

```
#define MAX 1000
struct eventos {
   char ocorrencia[200];
   char dataHora[50];
};
struct eventos eve[MAX];
int inicio = 0;
int fim = 0;
int processaEvento (struct eventos *recuperado) {
   if(inicio == fim) {
     return - 1;
   }
   else {
     inicio++;
     copiaEvento (recuperado, eve[inicio - 1]);
     return 0;
   }
}
int insereEvento (struct eventos *novo) {
   if (fim == MAX) {
      return -1;
   }
   else {
     copiaEvento (eve[fim], novo);
     fim++;
     return 0;
   }
}
```

É correto afirmar que a estrutura de dados e a funcionalidade desse código tratam-se de

( ) Uma fila que processa os eventos de acordo com seu respectivo grau de prioridade.
( ) Uma pilha que processa primeiro os eventos mais recentes.
( ) Uma pilha que processa os eventos na ordem escolhida pelo operador.
(X) uma fila que processa primeiro os eventos mais antigos. [ALTERNATIVA CORRETA]
( ) Uma pilha que processa primeiro os eventos mais antigos.

2-A área de complexidade de algoritmos abrange a medição da eficiência de um algoritmo frente à quantidade de operações realizadas até que ele encontre seu resultado final.A respeito desse contexto, suponha que um arquivo texto contenha o nome de N cidades de determinado estado, que cada nome de cidade esteja separado do seguinte por um caracter especial de fim de linha e classificado em ordem alfabética crescente. Considere um programa que realize a leitura linha a linha desse arquivo, à procura de nome de cidade

Com base nessa descrição, verifica-se que a complexidade desse programa

Escolha uma:
a. O(1), em caso de busca sequencial.
b. O(N), em caso de busca sequencial.
c. O(log2N)O(log2N), em caso de busca binária.
d. O(N), em caso de transferência dos nomes para uma árvore binária e, então, realizar a busca.
e. O(log2N)O(log2N), em caso de transferência dos nomes para uma árvore binária e, então, realizar a busca.

3-

Considere o algoritmo que implementa o seguinte processo: uma coleção desordenada de elementos é dividida em duas metades e cada metade é utilizada como argumento para a reaplicação recursiva do procedimento. Os resultados das duas reaplicações são, então, combinados pela intercalação dos elementos de ambas, resultando em uma coleção ordenada. Qual é a complexidade desse algoritmo?

Escolha uma opção:
- a. $O(n \times \log n)$
- b. $O(n^2)$
- c. $O(\log n \times \log n)$
- d. $O(2^n)$
- e. $O(n^{2n})$

4- A) Insere Inicio na Lista Duplamente Encadeada.

```
void Lista::insere_inicio(int i)
{
```

```
        no_dado* p;
        no_dado* novo = new no_dado(i, this->cabeca);
        if (this->isEmpty(this->cabeca))
        {
                this->cabeca = novo; //cabeca = novo
                this->cauda = novo;
        }
        else
        {
                novo->prox = this->cabeca;
                novo->prox->ant = novo;
                this->cabeca = novo;
        }
}
```

B) Insere Fim na Lista Duplamente Encadeada.

```
void Lista::insere_fim(int i)
{
        no_dado* p;
        no_dado* novo = new no_dado(i, this->cabeca);
        if (this->isEmpty(this->cabeca))
        {
                this->cabeca = novo; //cabeca = novo
                this->cauda = novo;
        }
        else{
                novo->ant->prox = novo;
                novo->ant = this->cauda;
                this->cauda = novo;
        }
}
```

5- Propor e implementar uma melhoria no código de BubbleSort.

Proposta: para melhorar a eficiência do código, poderia implementar uma condicional que verifique se o vetor já está ordenado, ou seja, caso não houvesse a troca entre os valores, o código poderia ser interrompido, poupando tempo de execução.

Código:

Faz ai otario