PCC534 – Análise de Algoritmos e Estruturas de Dados

Atividade Prática 1 – Listas, Pilhas e Filas

Nesta atividade, deverão ser entregues implementações de algoritmos para estruturas de dados de listas, pilhas e filas e um relatório com análise das questões levantadas. Os enunciados e exemplos estão em linguagem C. Entretanto, as implementações podem ser feitas em linguagem C, C++, Java ou Python, desde que sejam utilizados recursos equivalentes para implementação dos elementos básicos das estruturas de dados, sem utilização de bibliotecas ou recursos das linguagens prontos que já disponibilizem os elementos solicitados nas implementações. Os arquivos com as implementações e relatório deverão ser compactados em um único arquivo a ser entregue no Campus Virtual.

1) Na primeira parte da implementação, você deverá implementar duas versões de lista encadeada dinâmica, com ponteiro para identificação da cauda e sem ponteiro para identificação da cauda.

O primeiro tipo de lista deverá ser definido da seguinte forma:

```
typedef struct elemento* Lista1;

struct elemento{
      int dado;
      struct elemento *prox;
};
```

Implemente as seguintes funções para criação da lista, inserção na cabeça da lista (primeira posição) e na cauda (última posição), para o tipo Lista1:

Lista1* criaLista() – deve alocar e retornar um ponteiro para uma nova lista

void liberaLista(Lista1* li) – limpa a lista e desaloca os espaços de memória

insereNaCabeca(int dado, Lista1* li) – Insere um elemento na primeira posição da lista

insereNaCauda(int dado, Lista1* li) – Insere um elemento na última posição da lista

O segundo tipo de lista deverá ser definido da seguinte forma:

```
typedef struct lista2{
      struct elemento *cabeca;
      struct elemento *cauda;
} Lista2;

struct elemento{
      int dado;
      struct elemento *prox;
};
```

Implemente as seguintes funções para criação da lista, inserção na cabeça da lista (primeira posição) e na cauda (última posição), para o tipo Lista2:

Lista2* criaLista() – deve alocar e retornar um ponteiro para uma nova lista

void liberaLista(Lista2* li) – limpa a lista e desaloca os espaços de memória

insereNaCabeca(int dado, Lista2* li) – Insere um elemento na primeira posição da lista

insereNaCauda(int dado, Lista2* li) – Insere um elemento na última posição da lista

2) Crie uma definição de TAD tipo Pilha e uma função inserirNaPilha(int dado, Pilha *pilha). Sua implementação de pilha deverá utilizar como base uma Lista. Qual tipo de Lista seria mais adequado para implementar a inserção na pilha, a TAD Lista1 ou a TAD Lista2?

3) Crie duas versões de uma TAD Fila, cada uma delas com uma função inserirNaFila(int dado, Fila *fila). Uma versão deverá utilizar como base a TAD Lista1, e a outra versão deverá utilizar como base a TAD Fila2.

   Analise qual seria a complexidade das duas versões da função inserirNaFila(int dado, Fila *fila), e dê qual seria a função f(n) que indica a complexidade de pior caso de *O(f(n))* para as duas versões de implementação de fila. Justifique em seu relatório a forma como você realizou esta análise.