Programação III
Lista de exercícios : Listas encadeadas parte 1

Aluno: Thiago Pedro Ferreira de Moraes

1 É possível implementar a operação de conjuntos dinâmicos Inserir em uma lista singularmente encadeada em tempo constante? E a operação Remover? (Obs: escrever os algoritmos)

A operação Inserir de conjuntos dinâmicos é possível de ser implementada em uma lista singularmente encadeada, em tempo constante. Para isso, é preciso "reapontar"a cabeça ao novo elemento da lista - o qual apontará, também, para o próximo elemento -, após desvinculá-la do antigo primeiro elemento. Como isso independe do tamanho da lista, a implementação é em tempo constante.

```
INSERIR(L,x)
    x.proximo =  L.cabeca  // t=c_2
    L.cabeca = x  // t=c_1

A=c_1+c_2

Seja x o tamanho da lista e t, o tempo de execução:
t(x)= A

Concluindo que o código roda em tempo constante.
```

A operação remover ocorre da mesma forma. A cabeça da lista encadeada é reapontada para o próximo elemento da lista, verificando antes a existência deste. Esta operação também ocorre em tempo constante.

```
REMOVER(L,x)
    se x.proximo != NIL // c_1
        L.cabeca = x.proximo  // c_2
    senão L.cabeca = NIL // c_3
```

```
A=c_1+c_2
B= c_3

Seja x o tamanho da lista e t, o tempo de execução:
t(x)= max(A,B)

Concluindo que o código roda em tempo constante.
```

2 Implemente uma pilha usando uma lista singularmente encadeada L. As operações Push e Pop devem rodar em tempo constante.

```
PUSH(L,x)
    x.proximo =  L.cabeca
    L.cabeca = x

POP(L,x)
    se x.proximo != NIL
        L.cabeca = x.proximo
    senão L.cabeca = NIL
```

3 Implemente uma fila usando uma lista singularmente encadeada L. As operações Enfileirar e Desenfileirar devem rodar em tempo constante.

Para implementar uma fila usando uma lista encadeada, foi utilizado o endereco do último elemento da lista (L.ultimo). Desta forma, todo novo elemento ocupa o final da fila. Ao desenfileirar, a cabeça da lista simplesmente aponta para o segundo elemento, tornando este o novo primeiro elemento. Seguem os códigos:

```
ENQUEUE(L,x)
    se L.cabeca != NIL
        L.ultimo.proximo = x
        L.ultimo = x
    senão
        L.cabeca = x
        L.ultimo = x


DEQUEUE(L,x)
    se x.proximo != NIL
        L.cabeca = x.proximo
    senão L.cabeca = NIL
```