

Algoritmos e Complexidade

LEI (2º ano)

Perguntas do 1º Mini-Teste

Ano Lectivo de 2011/12

Considere as seguintes definições de tipos em C, para implementação de uma lista ligada. Cada nó da lista inclui uma chave (um número inteiro) e uma referência (apontador) para informação associada à chave (neste caso uma estrutura com dois campos).

```
typedef struct dados {
    int x;
    int y;
} Dados;

typedef struct node {
    int chave;
    Dados *info;
    struct node *next;
} Node;

typedef struct listal {
    int comprimento;
    Node *lista;
} ListaL;
```

1. Apresente uma implementação da seguinte função, que deve inserir um par chave/valor na última posição da lista. Note que:
 - a função deve devolver informação actualizada sobre a lista, com o comprimento actualizado.
 - os dados associados à chave são passados por valor, pelo que a função deverá ela própria reservar o espaço de memória necessário ao seu armazenamento.

```
ListaL insereFim(ListaL lista, int chave, Dados valor);
```

2. Apresente uma implementação da seguinte função, que deve inserir um par chave/valor na última posição da lista. Note que:
 - A função deve devolver informação actualizada sobre a lista, com o comprimento actualizado. Note que a lista é passada por referência (apontador).
 - Os dados associados à chave são passados através de um apontador que pode ser directamente armazenado na lista, pelo que a função não deverá ela própria reservar o espaço de memória necessário ao seu armazenamento.

```
void insereFim(ListaL *lista, int chave, Dados *valor);
```

3. Apresente uma implementação da seguinte função, que deve substituir os dados associados a uma determinada chave na lista. A função deverá retornar 0 no caso de a operação não ter sucesso, e 1 no caso contrário. Neste último caso a função deverá retornar a posição em que a chave foi encontrada numa variável passada por referência.

```
int substituiDados(ListaL lista, int chave, Dados valor, int *posicao);
```

4. Apresente uma implementação da seguinte função, que deve substituir os dados associados a uma determinada chave na lista. A função deverá retornar 0 no caso de a operação não ter sucesso, e 1 no caso contrário. Neste último caso a função deverá retornar o valor original dos dados associados à chave na variável `valor`, passada por referência.

```
int substituiDados(ListaL lista, int chave, Dados *valor);
```

5. Apresente uma implementação da seguinte função, que deve remover o nó correspondente a uma determinada chave da lista. Note que:

- A função deve devolver informação actualizada sobre a lista, com o comprimento actualizado.
- A função deve devolver os dados associados à chave removida na variável passada por referência.

```
ListaL Remove(ListaL lista, int chave, Dados *valor);
```

6. Apresente uma implementação da seguinte função, que deve remover o nó correspondente a uma determinada chave da lista. Note que:

- A função deve devolver informação actualizada sobre a lista, com o comprimento actualizado. Por esse motivo, a lista é passada por referência.
- A função deve retornar os dados associados à chave removida.

```
Dados Remove(ListaL *lista, int chave);
```

7. Apresente uma implementação da seguinte função, que deve substituir o par chave/valor armazenado numa determinada posição da lista. A função deverá retornar 0 no caso de a operação não ter sucesso, e 1 no caso contrário. Neste último caso a função deverá retornar o par chave/dados substituídos, utilizando as variáveis passadas por referência.

```
int substituiDados(ListaL lista, int *chave, Dados *valor, int posicao);
```

8. Apresente uma implementação da seguinte função, que deve remover o par chave/valor armazenado numa determinada posição da lista. A função deverá retornar informação actualizada sobre a lista, incluindo o comprimento depois da remoção.

```
ListaL removePos(ListaL lista, int posicao);
```