Lista 6 - ED

Wellington Silva
Tiago Barradas
Escola de Matemática Aplicada, Fundação Getulio Vargas, Brasil.

May 31, 2023

Considerações: Submissão do código em um aquivo `.cpp` ou um zip com um `.cpp` por questão.

Muito parecida com a lista encadeada, mas com uma diferença crucial: o fim da lista, ao invés de ter um *nullptr*, temos um ponteiro para o primeiro nó da lista. O nó é o mesmo da lista encadeada:

```cpp
struct Node {
    int data;
    Node *next;
};
```

Problema 1: Anteriormente, implementamos para as outras estruturas a função insert, que adiciona um determinado elemento na lista. Dessa vez, assuma que a lista só pode receber inteiros, e implemente uma outra versão do insert, de forma que insira os elementos não no final da lista, e sim, em sua posição correta em ordem crescente (ou seja, em uma lista 1 -> 3 -> 7, ao inserir o número 5, essa função o inseriria entre o 3 e o 7). Não se esqueça que a lista deve ser circular.

Problema 2: A lista circular e a lista encadeada são bem parecidas, faça uma função de verificação que, dado um ponteiro para o primeiro elemento de uma lista, verifique (retornando um bool) se a lista dada é circular ou não.

Problema 3: Dada uma lista encadeada, suponha que queremos fazer uma rotação de k elementos nessa lista (por exemplo, uma rotação de 2 elementos transforma "1 -> 4 -> 3 -> 8 -> 2 -> nullptr" em "3 -> 8 -> 2 -> 1 -> 4 -> nullptr".). Crie uma função que realize essa operação, usando o conceito de lista circular, e retornando um ponteiro que apontará para a nova head da lista encadeada, sem criar uma lista nova.

Problema 4: Inspirado no problema dos soldados apresentado na questão 2 da lista 2. Construa uma lista circular com os nós da seguinte forma:

```cpp
struct Node {
    int id;
    bool is_alive;
    Node *next;
};
```

Com n soldados numerados de 1 a n, e as variáveis "is_alive" sendo True. Depois, use essa lista para criar uma função que irá resolver o mesmo problema da lista 2, do soldado Josefo (https://en.wikipedia.org/wiki/Josephus_problem), de forma geral, recebendo o número de soldados que irão se matar e $k \geq 2$ indicando qual soldado será morto, começando sempre do 1º. (para $k = 2$ o 1 mata o 2, o 3 mata o 4, o 5 mata o 6, ..., para $k = 3$ o 1 mata o 3, o 4 mata o 6, o 7 mata o 9,... e segue).