

Lista 4 - ED (Linked List)

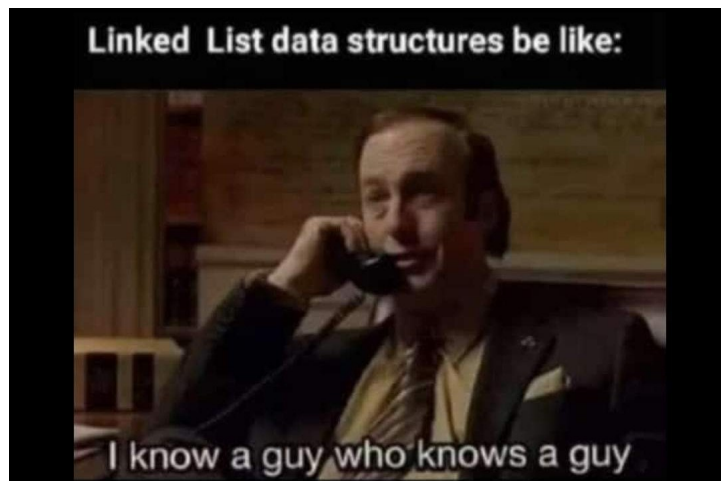
Wellington Silva

Tiago Barradas

Escola de Matemática Aplicada, Fundação Getulio Vargas, Brasil.

May 24, 2023

Entrega: Submissão do código em um arquivo .cpp ou em um por questão. Você pode reusar as funções implementadas na prova A1 (vide https://github.com/barrafas/Monitoria_ED/blob/main/A1_gabarito.cpp).



Problema 1: Faça uma função equivalente a `sum(.)` em python para linked list. A partir da head de uma linked list, ela retorna a soma dos elementos. Como exemplo, o seguinte código deve retornar 55:

```
1 #include <iostream>
2
3 struct Node {
4     int data;
5     Node *next;
6 };
7
8 void insert(Node *head, int data) {
9     Node *newNode = (Node*) malloc( sizeof(Node));
10    newNode->data = data;
11    newNode->next = nullptr;
12
13    Node *current = head;
14    while (current->next != nullptr) {
15        current = current->next;
16    }
17    current->next = newNode;
18 }
19
```

```

20 void printList(Node *head) {
21     Node *current = head;
22     while (current != nullptr) {
23         std::cout << current->data << " ";
24         current = current->next;
25     }
26     std::cout << std::endl;
27 }
28
29 // your sum function
30
31 int main(){
32     Node *head = (Node*) malloc(sizeof(Node));
33     head->data = 1;
34     for (int i = 1; i < 10; i++) {
35         insert(head, 1+i);
36     }
37
38
39     std::cout << sum(head);
40 }

```

Problema 2: Aproveitando a estrutura anterior, faça uma função que dadas 2 linked lists, cria uma nova linked list, (não deve modificar as já existentes) que é a concatenação das duas. Como exemplo a seguinte `main` deverá printar a sequência 1, 2, 4, ..., 18, 1, 5, 10, ..., 45

```

1 int main(){
2     Node *head1 = (Node*) malloc(sizeof(Node));
3     head1->data = 1;
4     for (int i = 1; i < 10; i++) {
5         insert(head1, 2*i);
6     }
7
8     Node *head2 = (Node*) malloc(sizeof(Node));
9     head2->data = 1;
10    for (int i = 1; i < 10; i++) {
11        insert(head2, 5*i);
12    }
13
14
15    Node *head = concat(head1, head2);
16    printList(head);
17 }

```

Problema 3: Dado uma linked list como anteriormente, implemente uma função equivalente ao bubble sort modificando a lista existente. Como exemplo a seguinte `main` deverá printar a sequência -9, -7, -5, -3, -1, 1, 2, 4, 6, 8 (importe a biblioteca `cmath` para o exemplo).

```

1 int main(){
2     Node *head1 = (Node*) malloc(sizeof(Node));
3     head1->data = 1;
4     for (int i = 1; i < 10; i++) {
5         insert(head1, pow(-1, i) * i);
6     }

```

```
7
8     sort(head1);
9     printList(head1);
10 }
```

Problema 4: (Matriz?) A nossa linked list costuma usar apenas inteiros, mas isso nem sempre é a melhor opção. Aqui você deve implementar uma linked list de linked lists de inteiros. Para isso você deve implementar um `Node2` que em vez de receber um inteiro, deve receber um ponteiro para uma linked list. Com isso implemente as funções `insert2` (que insere uma linked list no fim) e `printList2` (que printa os elementos as linked lists internas). Observe o exemplo de main:

```
1 int main(){
2     Node *head1 = (Node*) malloc(sizeof(Node));
3     head1->data = 1;
4     for (int i = 1; i < 10; i++) {
5         insert(head1, pow(-1, i) * i);
6     }
7
8     Node *head2 = (Node*) malloc(sizeof(Node));
9     head2->data = 1;
10    for (int i = 1; i < 10; i++) {
11        insert(head2, 2 * i);
12    }
13
14    Node2 *head = (Node2*) malloc(sizeof(Node2));
15    head->node = head1;
16    head->next = nullptr;
17
18    insert2(head, head2);
19
20    printList2(head);
21 }
```