Linked List
(Programming Club 1)

Tomasz Kosciuszko

01.10.2018

1 Introduction

Linked list is a simple way to store data. The advantage of it is the fact that it does not ask you to know in advance how much memory you are going to need. You can insert elements one-by-one in a clean way.

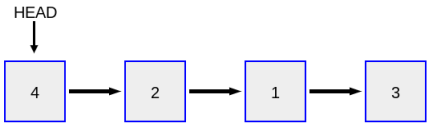

Figure 1: You guess what this is

2 How it works

The structure is built out of n "cells" with some meaningful data (above numbers). Each cell points towards a next one, most often keeping the memory address of it. Like here in C++:

```
class Cell {
 public:
    int number;
    Cell* next;
};
```

Now it is enough if the object that represents the whole list keeps track of just the first element:

```
class LinkedList {
 public:
    Cell* first_cell;
};
```

What operations can we perform with a given list? Let's try:

```
void Insert(LinkedList* list, int number) {  // So easy!
    Cell new_cell;
    new_cell.number = number;
    new_cell.next = list->first_cell;
    list->first_cell = &new_cell;
}

void GetLength(LinkedList* list) {  // So slow!
    int length = 0;
    Cell* cell = list->first_cell;
    while (cell != null) {
        length++;
        cell = cell->next;
    }
    return length;
}
```

That's it! Not the most complicated data structure in the universe. But think about the following:

- What happens if somebody introduces a loop in your list and then calls the above "GetLength" function?
- Is it possible to detect if a linked list contains a loop? Try to write an "IsLoop" function. It is very tricky!
- Can you improve the efficiency of the "GetLength" function? Perhaps you can store length in the "LinkedList" class?
- Can you write a function "Remove" that will take an int and remove all copies of it from the list?

Coding here!

- Have a look at implementation of Linked List in your favourite language and solve a problem involving linked lists: https://tinyurl.com/ydy2htdw
- Try to solve a **hard** problem: https://tinyurl.com/yctjcntf If you need inspiration search for "disjoint set data structure" in your favourite web search engine!