

Practice7

March 10, 2020

Mayra Cristina Berrones Reyes

1 Introduction

1.1 Arrays vs lists

Arrays and lists often are confused on any programming language. In our case, we searched for the concept of arrays and lists inside a python language. In python, this two structures have similar functions, as in both of them store data, but the main difference can be seen on their structures and how the data can be accessed.

A list in python is a data structure that holds a collection of items. They are declared with enclosed brackets like `[item1]`, and they appear in a specific order, that enables us to use an index to access the information. Lists are mutable, which mean they can change, add, or remove items after we create de list. The elements inside a list do not require to be unique, and a list can hold several different types of data inside (integers, strings, objects, etc.).

An array is also a data structure that stores a collection of items, they can be mutable, ordered, enclosed in brackets and have non unique items. The first thing that separates lists form arrays is that arrays can not hold different types of data.

Another difference is that arrays need to be declared in python, and lists do not. Python needs to import the library *Numpy* to be able to use an array. In Table 1 we can see the main differences between arrays and lists.

Taking into consideration the pros and cons of using arrays or lists, we demonstrate two problems to see if this features are correct.

Table 1: Main differences of lists and arrays in python.

Lists	Arrays
<ul style="list-style-type: none"> • Lists can be flexible and hold arbitrary data. • They are a part of the python syntaxes so they do not need to be declared first. • Can be resized quickly in a time efficient manner. Python initializes some elements in the list at the initialization. • Lists can hold different types of data. • Mathematical functions can not be directly applied to all the list, it needs to be applied to each item individually. • They consume more memory as they are allocated a few extra elements to allow for quicker appending. • To search for an item, you need to start from the first element, and go through all of the other items until you reach the one you want. • Delete and insertion are easy. 	<ul style="list-style-type: none"> • Arrays need to be imported from other libraries. • Arrays can not be resized, it needs to be copied to another larger array. • Arrays can only store items with values of uniform data types. • They are specially optimized for arithmetic computations. • Since arrays stay the size that they were initialized with, they are compact • In the array you can find an item easily by their index. • Operations like delete and insertion take a lot of computational time.

1.2 Palindrome

For this first example we have the problem of a palindrome. Remembering the differences we described in Table 1 we have that arrays are more well suited for this type of problem, since we can search and compare the contents of the first and last item on the array using the the index numbers, as we can see in Figure ?? . Then in Figure ?? we can see that we will have to pass the list several times to check between the first and the last items on the list, since each element is pointing to the direction were the next item is stored (it is not a fixed parameter).

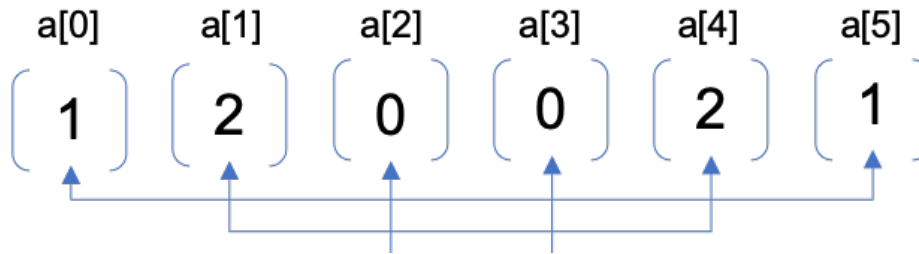


Figure 1: Palindrome example for an array

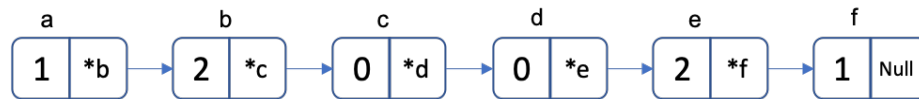


Figure 2: Palindrome example of a list

1.3 Delete all the duplicated values

Now we have another example in which we need to remove the duplicated values of an array. In this case, the array will not do so well, since we have to move all of the elements so that there is not a null value inside the array, and we can safely remove the last items that are null, as we can see on Figure ?? . On the other hand, we have the list, that on this case perform better than the arrays, because we can easily bypass the elements to remove, and have the remaining items point only to the unique values. This example can be seen in Figure ?? .



Figure 3: Example for deleting duplicated values of an array

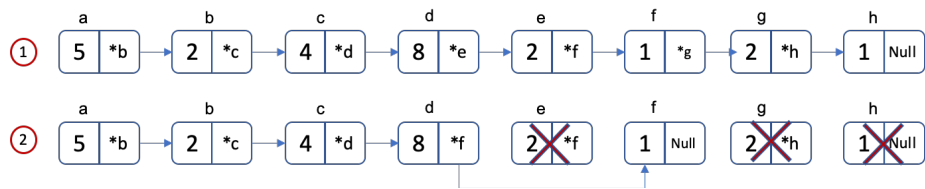


Figure 4: Example to delete duplicated values on a list

2 Conclusions

It took me a while to grasp the concept of this practice, because I am very used to handle python, and every time I saw something referenced as a list or an array, I thought they meant the same. Now, after reading more about the rules, I figured that I have been using lists too liberally, and that it may not seem very significant the computational time, because despite their drawbacks, both structures perform very fast, but I realize now that some of the problems I had in the past, programing simple things that resulted in errors or leaked memory could have been because I was not using the proper structure to store my data.