Arrays

A collection of elements / values eache identified by an array index or key

- index starts at zero (python, java, c++)
- because of the indexes: random access is possible

Multidimensional Arrays

It can prive to be very important in mathematical related computations (matrices)

Resume

1. Arrays are data structures in order to store items of the same type
2. We use indices are keys
3. Arrays can have as many dimensions as we want: one or two dimensional arrays are quite popular
4. For example: storing a matrix -> two dimensional array
5. Dynamic array: when the size of the array is changing dynamically

6. Applications: lookup table / hashtables, heaps

Advantages

- We can use random access because of the keys: `getItem(int index)` // O(1)
- Very easy to implement and to use
- Very fast data structure
- We should use arrays in applocation when we want to add items over and over again and we want to take items with given indexes

Disadvantages

- We have to know the size of the array at complile-time: so it is not so dynamic data structure
- If it is full: we have to create a bigger array and have to copy the values one by one // reconstructing an array is O(N) operation
- It is not able to store items with different types

Arrays Operation: Add Item

We can keep adding values to the array as far as the array is not full.

Time complexity: O(1).

```
Array.append("x")
```

Add Item with given index

We would like to insert a given value with a given index, if the index is empty.

Time complexity: O(N).

```
array[3] = 23
```

If the index isn’t empty we have to allocate items, or delete the item.

Remove Items

We would like to remove the last item, it is very simple, just remove it

Time complexity: O(1).

```
array = array[:-1]
```

Remove Items with given index

We would like to remove a value with a given index, it is not tha simple, we may have to shift items

Time complexity: O(N)

Bruna Santos - January 22, 2018 1:47 pm