dynamic_array

April 12, 2020

# 1 Dynamic Array - Amortized Cost

[1]:

```
import random as rd
import ctypes
```

[2]:

```
class DArray(object):
    def __init__(self):
        self._n = 0
        self._capacity = 1
        self._array = self.__create_low_level_array(self._capacity)

    def __len__(self):return self._n

    def __getitem__(self,i):
        if 0 <= i < self._n: return self._array[i]
        else: raise IndexError("index out of range")

    def __resize_array(self,capacity):                            # private␣
 ↪method
        new_array = self.__create_low_level_array(capacity)
        for i in range(self._n): new_array[i] = self._array[i]
        self._array = new_array
        self._capacity = capacity

    def __create_low_level_array(self,capacity):                  # private␣
 ↪method, C array tipinde dizi oluşturur
        py_obj_array = capacity * ctypes.py_object
        return py_obj_array()

    def append(self,obj):
        if self._n == self._capacity: self.__resize_array(2*self._capacity)   ␣
 ↪# array kapasitesi iki katını alır
        self._array[self._n] = obj
        self._n += 1

    def remove(self):                                              # array boş␣
 ↪değilse ancak eleman silinecek
```

```
        if self._n > 0:
            self._array[self._n] = None
            self._n -= 1
```

```
[3]: darray = DArray()
```

```
[4]: darray.append(rd.randint(0,100))
     darray.append(rd.randint(0,100))
     darray.append(rd.randint(0,100))
     darray.append(rd.randint(0,100))
     darray.append(rd.randint(0,100))
```

```
[5]: print("DArray[",end=" ")
     for i in darray: print(i,end=" ")
     print("]")
```

```
DArray[ 51 54 84 21 49 ]
```

```
[6]: darray.remove()
```

```
[7]: print("DArray[",end=" ")
     for i in darray: print(i,end=" ")
     print("]")
```

```
DArray[ 51 54 84 21 ]
```