# Complexitatea functiei „sortare\_nume()”

def sortare\_nume(*self*):  
 lista\_clienti = []  
 for el in *self*.\_\_repo\_inchiriere.afisare():  
 client = *self*.\_\_service\_clienti.cauta\_client\_dupa\_id(el.get\_client())  
 if client.get\_nume\_client() not in lista\_clienti:  
 lista\_clienti.append(client.get\_nume\_client())  
 lista\_clienti.sort()  
 return lista\_clienti

Caz favorabil = Caz defavorabil = Caz mediu

Complexitatea este O(n), complexitate liniara, unde n reprezintă numărul de clienți.

# Complexitatea functiei „bubble\_sort()”

def bubble\_sort(*list*, *key* = lambda *x*: x, *cmp* = lambda *x*, *y*: x>y, *reverse* = False):  
 '''  
 Functie de sortare folosind metoda bulelor  
 :param list: lista ce trebuie sortata  
 :param key: cheia dupa care se sorteaza  
 :param cmp: functia de comparare  
 :param reverse: True sau False daca lista trebuie sa fie sortata descrescator sau crescator  
 :return: -  
 '''  
 ok = False  
 while ok is False:  
 ok = True  
 for index in range(len(*list*)-1):  
 if *cmp*(*key*(*list*[index]), *key*(*list*[index+1])):  
 *list*[index], *list*[index+1] = *list*[index+1], *list*[index]  
 ok = False  
 if *reverse* == True:  
 *list*.reverse()

Cazul favorabil este cel in care lista este sortata. - **ϴ** (n)

Cazul defavorabil este cel in care lista este sortata descrescator – **ϴ** (n^2)

Cazul mediu este **O** (n^2) – Complexitate generala