Претраге и сортирање

Секвенцијално претраживање листа

Најједноставнији начин да се утврди да ли постоји тражени елемент у листи је испитивање сваког појединачног елемента из листе да ли је идентичан са траженим елементом.
Претрага се завршава када се пронађе по први пут тражени елемент (пронађе се његова позиција у листи) или када се утврди колико има у листи идентичних елементата са траженим елементом.
Овакав поступак се назива секвенцијално претраживање.
Секвенцијално претраживање се обавља над листом елемената који не морају обавезно бити сортирани.

0257 Секвенцијална претрага

```
def main():
    lista = unos_liste()
    prikaz_liste(lista)
    element = unos_elementa_za_pretragu()
    pronadjen = pretraga(lista, element)
    prikaz_rezultata(lista, element, pronadjen)

def unos_liste():
    print("Uneti u listu samo cele brojeve.")
    A = []
    jos = "da"
    while jos == "da":
        x = int(input("Uneti zeljeni broj u listu:"))
        A.append(x)
        print("Da li treba uneti jos jedan element u listu? ")
        jos = input("Uneti da ili bilo sta drugo za ne: ")
    return A

def prikaz_liste(A):
    print(A)

def unos_elementa_za_pretragu():
    a = int(input("Koji element se trazi u listi? "))
    return a

def pretraga(A, a):
    for x in range(len(A)):
        if A[x] == a:
            return True
    return False

def prikaz_rezultata(A, a, ima):
    print("U listi ", str(A))
    print("trazen je element ", a)
```

```
    if ima:
        print("Trazeni element se nalazi u listi.")
    else:
        print("Trazeni element se ne nalazi u listi.")

main()
```

Приметити да се претрага траженог елемента по листи завршава пре достизања краја листе.
Употребом секвенцијалне претраге се може прекинути претрага у било којем моменту претраге ако је постигнут услов за излазак из петље претраге.Овакав метод претраге није ефикасан јер се може десити да је на последњем месту у листи налази тражени елемент, што потенцијално повећава број потребних испитивања једнакости.

Број појављивања елемената у листи

Додатак у алгоритму секвенцијалне претраге је проналажење броја појављивања траженог елемента.
У овом алгоритму се увек долази до краја листе елемената пошто се мора открити број појављивања елемента.

0258 Секвенцијално бројање елемената

```
def pretraga(A, a):
    b = 0
    for x in range(len(A)):
        if A[x] == a:
            b += 1
    return b
```

Промењива b преноси у главни део кода број појављивања траженог броја у листи.
То значи да број појављивања може бити 0 или било који цео позитиван број.
Остатак функција се не мора мењати у односу на претходни пример, осим функције prikaz_rezultata().

0259 Претрага елемента у сортираној листи

```
def pretraga(A, a) :
    for i in range(len(A)) :
        if A[i] == a :
            return True
        elif A[i] > a :
            return False
    return False
```

У примеру се испитују три могућности: 1) Тражени број постоји у листи 2) Сви бројеви су већи од траженог броја а тражени број није у листи 3) Сви бројеви су мањи од траженог броја а тражени број није у листи

0260 Претрага најмањег елемента у листи

```
def pretraga(A):
    najmanji = A[0]
    for i in range(1, len(A)) :
       if A[i] < najmanji :
           najmanji = A[i]
    return najmanji
```

Бинарна претрага сортиране листе

0261 Бинарна претрага

```
def main():
    lista = [-34, -15, -9, -1, 0, 4, 5, 10, 13, 189, 1020]
    element = unos_elementa_za_pretragu()
    pronadjen = binarna_pretraga(lista, element)
    prikaz_rezultata(lista, element, pronadjen)

def unos_elementa_za_pretragu():
    a = int(input("Koji element se trazi u listi? "))
    return a

def binarna_pretraga(A, a):
    nizi = 0
    visi = len(A) - 1
    while nizi <= visi:
        sredina = (visi + nizi) // 2
        if A[sredina] == a:
            return True
        elif a < A[sredina]:
            visi = sredina - 1
        else:
            nizi = sredina + 1

    return False

def prikaz_rezultata(A, a, ima):
    print("U listi ", str(A))
    print("trazen je element ", a)
    if ima:
        print("Trazeni element se nalazi u listi.")
    else:
        print("Trazeni element se ne nalazi u listi.")

main()
```

После израчунавања позиције првог, последњег и средњег елемента, прво се испитује да ли је елемент на средњој позицији једнак траженом елементу.
Ако је то случај, одмах се враћа True тј пронађен је тражени елемент.
Ако није, рачуна се да ли је тражени елемент мањи од постојећег елемента на средњој позицији листе.
Ако јесте, модификује се вредност промењиве visi као (`visi = sredina - 1`); а ако није модификује се вредност промењиве visi као (`nizi = sredina + 1`).
У следећој итерацији петље, само ће се узети у обзир за претрагу део секвенце између овако модификованих позиција за nizi и visi промењиве.
На тај начин се смањује број потребних претрага да би се добио резултат претраге што повећава ефикасност кода.
Процес се понавља све док се елемент не пронађе или све док промењива nizi не постане већа од промењиве visi.
То ће се десити када не остане ни један елемент за обраду у листи, чиме се указује да тражени елемент није у датој листи.