



ESKİŞEHİR OSMANGAZİ ÜNİVERSİTESİ

Mühendislik Mimarlık Fakültesi

İnşaat Mühendisliği Bölümü

E-Posta: atopcu@ogu.edu.tr

Web: <http://mmf2.ogu.edu.tr/atopcu>

Bilgisayar Destekli Nümerik Analiz

Ders notları 2014

Ahmet TOPÇU

42

PROGRAMLAR: Arama (Searching) metotları

- Sıralı(Sequential) arama
- İkili(Binary) arama
- Blok ikili(Jump) arama

42. PROGRAMLAR: Arama (searching) metotları

Sayısal arama:

Doğrusal arama(Sequential veya Linear search), İkili arama(Binary search) ve Blok ikili arama(Jump search) metotlarının test sonuçları ve programları aşağıda verilmiştir. Bu metotların hepsi de hemen aynı miktarda bellek kullanmaktadır. Gerçek sayılardan oluşan n elemanlı $\underline{a}(n)$ vektörü kullanılarak her metodun hızı belirlenmiştir. Aranılan sayının $\underline{a}(n)$ vektöründe bulunduğu yere bağlı olarak arama yöntemleri farklı davranış sergilerler. Bu nedenle testler

$\underline{a}(n)=[1, 2, 3, \dots, n-1, n]$ sıralı vektöründe

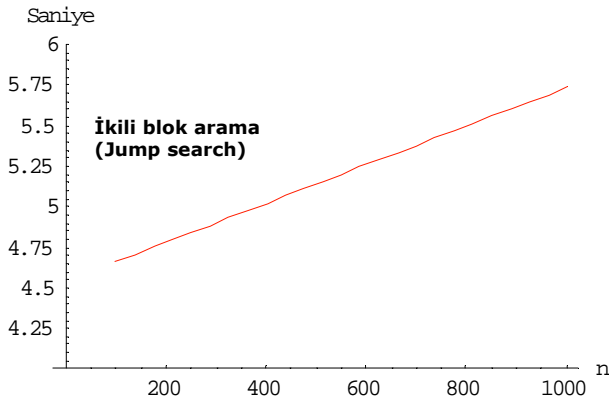
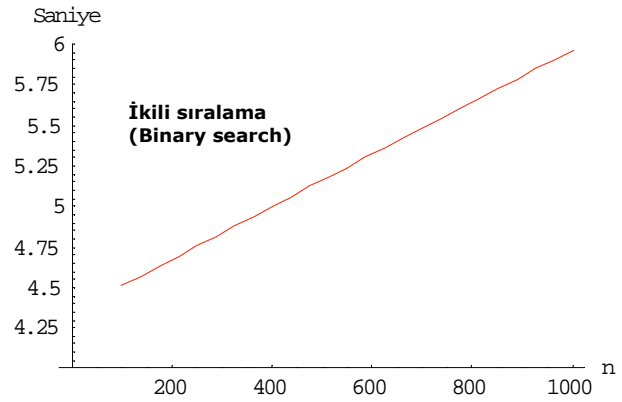
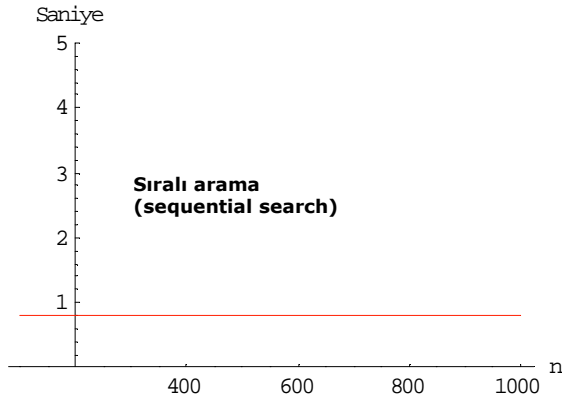
1 ve $n-1$ sayısı için yapılmıştır.

Eleman sayısı $n=100, 200, \dots, 1000$ alınarak $\underline{a}(n)$ vektörü kurulmuş, 1 veya $n-1$ sayısı 1000000 (bir milyon) kez aranmıştır. Aynı vektörde aynı sayının 1000000 kez aranmasının nedeni, ölçülebilir bir zaman elde etmeğe yöneliktir. Test sonuçlarının n -zaman(saniye) grafikleri izleyen sayfalarda verilmiştir.

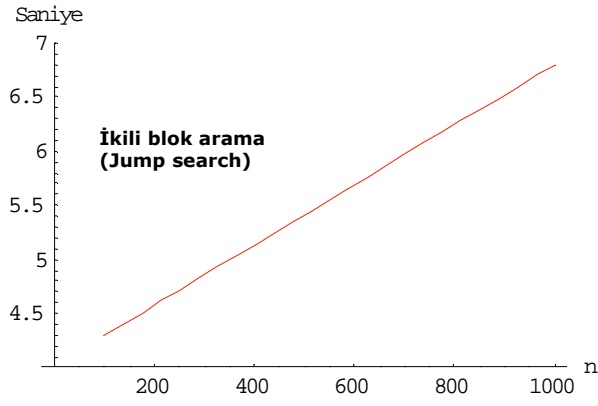
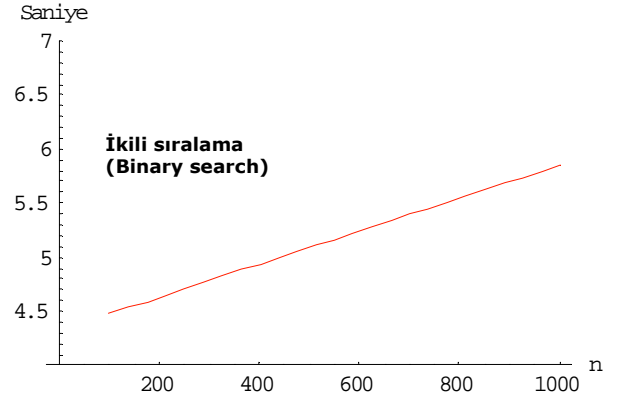
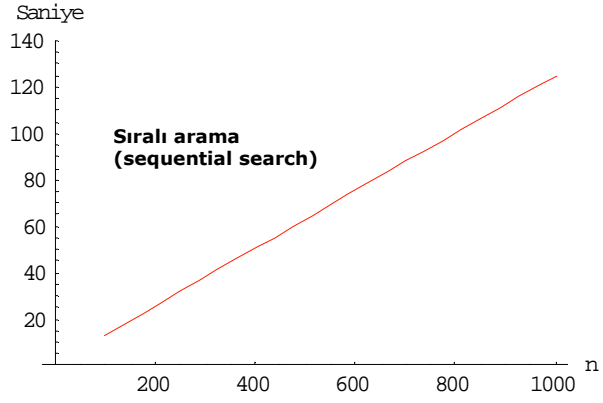
Programlarda $\underline{a}(n)$ vektörü değişkeni tek hassasiyetli(SINGLE) tanımlanmıştır. İstenirse Çift hassasiyetli veya başka bir tip olarak (DOUBLE, INTEGER,...) tanımlanabilir. Ayrıca $\underline{a}(n)$ vektörünün tamamının bellekte olduğu varsayılmıştır.

Karşılaştırma:

$\underline{a}=[1, 2, \dots, n-1, n]$ sayısal vektöründe 1 sayısının aranması:



$a=[1, 2, \dots, n-1, n]$ sayısal vektöründe $n-1$ sayısının aranması:



SONUÇ: En iyi arama metodu hangisidir?

1. Vektör sırasız ise ve sıralanmak istenmiyorsa: Sıralı arama kullanılır diğerleri kullanılamaz.
2. Vektör sıralı ise: İkili ve Blok ikili sıralama kullanılır. İkili arama tercih edilir. Zaman açısından ikili ve Blok ikili arama arasında hemen hiç fark yoktur. İkili aramanın programı daha basittir.

```

'-----Arama(Searching) test-----
' Ahmet TOPÇU, Anadolu Üniversitesi, Eskişehir, 1985
' Eskişehir Osmangazi Üniversitesi, 2010

' Sayısal vektörlerde arama yöntemleri:
' Sıralı arama(Sequential search), ikili arama(binary earch)

' a(n): Sayılar içeren vektör
' n:vektörün eleman sayısı
' Aranan: A(n) vektörü içinde aranacak değer
' index=0 aranan değer bulunamadı
' index>0 aranan değer index nolu satırda bulundu
'-----
DEFINT I-N
DEFSNG A-H, O-Z

DECLARE FUNCTION iSequentialSearch (n, a(), Aranan)
DECLARE FUNCTION iBinarySearch (n, a(), Aranan)
DECLARE FUNCTION JumpSearch (n, a(), Aranan)
DIM i AS LONG
DATA Sequential,Binary,Jump,Son

' n elemanlı a vektörü oluştur:
n = 10000
DIM s$(4), a(n)

FOR i = 1 TO n
a(i) = i
NEXT i

Aranan = n - 1' bu değer bulunduğ u satırı bul

100 CLS
RESTORE
FOR i = 1 TO 4
READ s$(i)
PRINT i; s$(i)
NEXT i

PRINT
INPUT "Seçenek="; iSecenek
IF iSecenek = 4 THEN END

t = TIMER
SELECT CASE iSecenek
CASE 1
FOR i = 1 TO 1000000
iNdex = iSequentialSearch(n, a(), Aranan)
NEXT i
CASE 2
FOR i = 1 TO 1000000
iNdex = iBinarySearch(n, a(), Aranan)
NEXT i
CASE 3
FOR i = 1 TO 1000000
iNdex = JumpSearch(n, a(), Aranan)
NEXT i
CASE 4
END
CASE ELSE
GOTO 100
END SELECT

Sure = TIMER - t
PRINT s$(iSecenek); " Arama: "
PRINT Aranan; " sayısının bulunduğu index="; iNdex
PRINT : PRINT "Süre="; Sure; "Saniye"
INPUT "Enter tuşuna basınız"; z$
GOTO 100

END ' Arama test sonu

```

Arama ana programı

iBinarySearch alt programı

```

FUNCTION iBinarySearch (n, a(), Aranan)
'-----
' a(n) sıralı vektöründe Aranan değerini arar

' veri:
' a(n): küçükten-büyük e sıralı sayılar içeren vektör
' Aranan: aranan değer

' çıktı:
' iBinarySearch=iNdex : Aranan değer iNdex nolu satırda bulundu
' iBinarySearch=0: Aranan değer bulunamadı
'-----
iStart = 0
iEnd = n
' iStart+1 ile iEnd arasında aranan değerini ara

WHILE iStart <= iEnd

iNdex = (iStart + iEnd) / 2
IF a(iNdex) = Aranan THEN
iBinarySearch = iNdex ' index nolu satırda bulundu
EXIT FUNCTION
END IF

IF a(iNdex) > Aranan THEN
iEnd = iNdex - 1
ELSE
iStart = iNdex + 1
END IF

WEND

iBinarySearch = 0 ' bulunamadı

END FUNCTION ' iBinarySearch sonu

```

```

FUNCTION JumpSearch (n, a(), Aranan)
'-----
' a(n) vektöründe Aranan değerini arar

'veri:
' a(n): sayılar içeren vektör(sıralı olması gerekmez)
' n: a vektörünün satır sayısı
' Aranan: aranan değer

'çikti:
' JumpSearch=index : Aranan değer index nolu satırda bulundu
' JumpSearch=0: Aranan değer bulunamadı
'-----

    LenBlok = n ^ (2 / 3)
    iBlokSayisi = INT(n / LenBlok)
    iStart = LenBlok
    iEnd = n

    iNdex = 0
    JumpSearch = 0
    FOR i = iStart TO iEnd STEP LenBlok
        IF a(i) >= Aranan THEN
            iNdex = i
            EXIT FOR
        END IF
    NEXT i

    IF iNdex = 0 THEN
        IF a(n) < Aranan THEN EXIT FUNCTION
        iStart = iBlokSayisi * LenBlok + 1
        ELSE
            IF a(iNdex) = Aranan THEN
                JumpSearch = iNdex
                EXIT FUNCTION
            END IF
            iStart = (INT(iNdex / LenBlok) - 1) * LenBlok + 1
            iEnd = iNdex
        END IF

' Block içinde binary search
        WHILE iStart <= iEnd
            iNdex = (iStart + iEnd) / 2
            IF a(iNdex) = Aranan THEN
                JumpSearch = iNdex
                EXIT FUNCTION
            END IF

            IF a(iNdex) > Aranan THEN
                iEnd = iNdex - 1
            ELSE
                iStart = iNdex + 1
            END IF
        WEND

        JumpSearch = 0
    END FUNCTION

```

JumpSearch alt programı

iSequentialSearch alt programı

```

FUNCTION iSequentialSearch (n, a(), Aranan)
'-----
' a(n) vektöründe Aranan değerini arar

'veri:
' a(n): sayılar içeren vektör(sıralı olması gerekmez)
' Aranan: aranan değer

'çikti:
' iBinarySearch=index : Aranan değer index nolu satırda bulundu
' iBinarySearch=0: Aranan değer bulunamadı
'-----

    index1 = 1
    index2 = n
' index1 ile index2 satırları arasında aranan değerini ara
    FOR iNdex = index1 TO index2
        IF a(iNdex) = Aranan THEN
            iSequentialSearch = iNdex ' index nolu satırda bulundu
            EXIT FUNCTION
        END IF
    NEXT iNdex
    iSequentialSearch = 0 ' bulunamadı

    END FUNCTION ' iSequentialSearch sonu

```