



ESKİŞEHİR OSMANGAZİ ÜNİVERSİTESİ

Mühendislik Mimarlık Fakültesi

İnşaat Mühendisliği Bölümü

E-Posta: ogu.ahmet.topcu@gmail.com

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

Bilgisayar Destekli Nümerik Analiz

Ders notları 2014

Ahmet TOPÇU

$$\underline{a} = \begin{bmatrix} 26.7 \\ 2.09 \\ -8.88 \\ 0.975 \end{bmatrix} \rightarrow \text{Sayısal sıralama} \rightarrow \underline{a} = \begin{bmatrix} -8.88 \\ 0.975 \\ 2.09 \\ 26.7 \end{bmatrix}$$

$$\underline{a} = \begin{bmatrix} İlhan \\ Zeynep \\ Çağla \\ Şirin \\ Işık \end{bmatrix} \rightarrow \text{Türkçe metin sıralama} \rightarrow \underline{a} = \begin{bmatrix} Çağla \\ Işık \\ İlhan \\ Şirin \\ Zeynep \end{bmatrix}$$

41

PROGRAMLAR: Sıralama (Sorting) metotları

- Bubble sıralama
- Insertion sıralama
- Shell sıralama
- Quick sıralama
- Gnome sıralama

41. PROGRAMLAR: Sıralama (sorting) metotları

Sayısal sıralama:

Bubble, Insertion, Heap, Shell, Quick ve Gnome sıralama metotlarının test sonuçları ve programları aşağıda verilmiştir. Bu metotların hepsi de hemen aynı miktarda bellek kullanmaktadır. Gerçek sayılardan oluşan n elemanlı $a(n)$ vektörü kullanılarak her metodun hızı belirlenmiştir. $a(n)$ vektörünün yapısına bağlı olarak sıralama yöntemleri çok değişik davranış göstermektedir. Tek bir vektör ile karşılaştırma yapmak yanlış sonuca götürür. Bu nedenle üç farklı yapıda $a(n)$ vektörü seçilmiştir. Aynı tip bir vektör küçükten-büyükçe tüm sıralama metotları ile ayrı ayrı sıralanarak her metodun hızı belirlenmiştir. Test için kullanılan $a(n)$ vektörleri aşağıdaki gibi seçilmiştir:

t tipi: $a(n)=[n, n-1, \dots, 2, 1]$ tamamen **ters** sıralı n tane ondalık sayı içeren vektör.

h tipi: $a(n)=[2,3, \dots, n-1, n, 1]$ **hemen hemen** sıralı n tane ondalık sayı içeren vektör(sadece 1 sayısı doğru yerinde değil).

r tipi: $a(n)=[\dots \text{rasgele sayılar} \dots]$ n tane **rasgele** ondalık sayı içeren sırasız vektör.

Eleman sayısı $n=100, 200, \dots, 1000$ alınarak $a(n)$ vektörü kurulmuş ve 1000 kez sıralanmıştır. Aynı vektörün 1000 kez sıralanmasının amacı, ölçülebilir bir zaman elde etmeye yöneliktir. Test sonuçlarının n -zaman(saniye) grafikleri izleyen sayfalarda verilmiştir.

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

Alfasayısal sıralama:

ÇçİİİİĞğÖöÜüŞş gibi Türkçeye özgü karakterlerin kod tablolarında mantıklı bir sıra izlememesi nedeniyle Türkçe metinlerin sıralanması özel kod ve program gerektirir. Örnek olması açısından, Quick sıralama yöntemi kullanılarak yazılmış bir program verilmiştir. Bu programda sadece

AaBbCcÇçDdEeFfGgĞğHhİiİjKkLlMmNnOoÖöPpQqRrSsŞşTtUuÜüVvWwXxYyZz0123456789

karakterleri dikkate alındığından, bunlar dışında (örneğin Â, α, ä, \$, {, @, gibi) karakterler içeren metinler doğru sıralanamayacaktır. Ancak; listede olmayan karakterler eklenerek program geliştirilebilir.

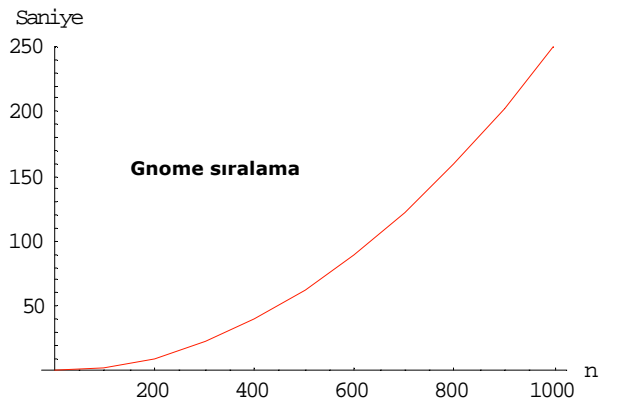
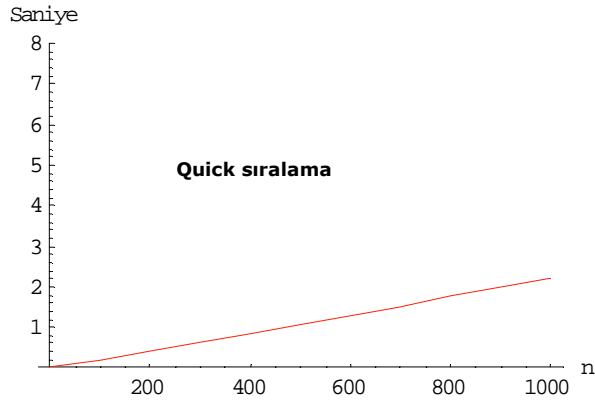
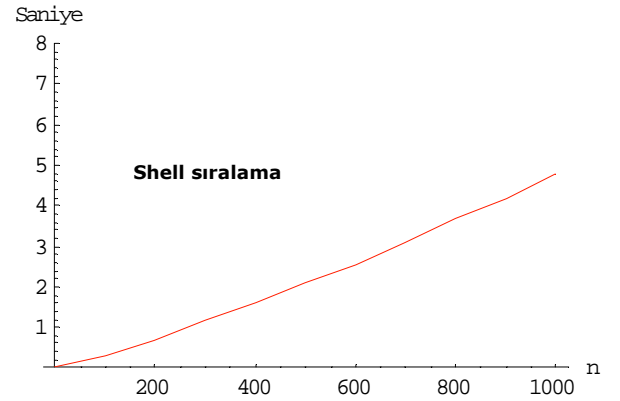
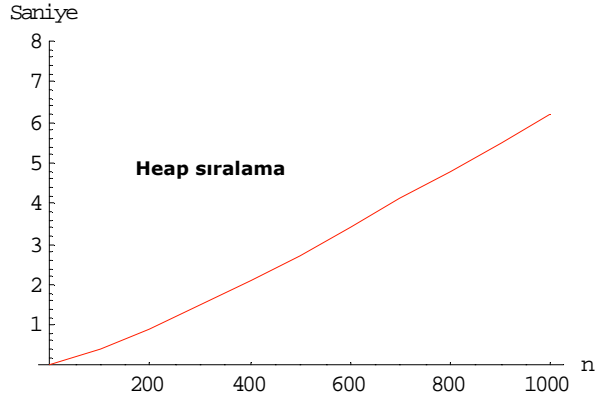
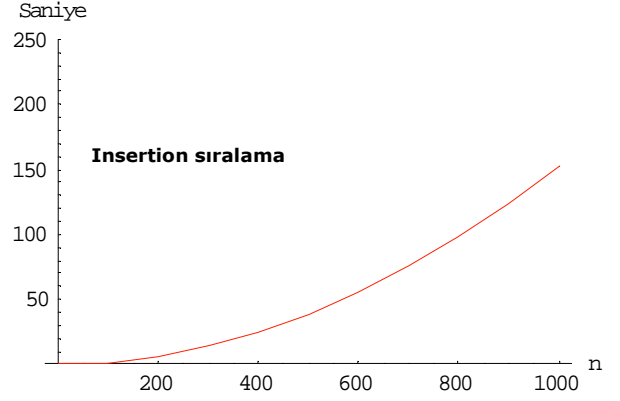
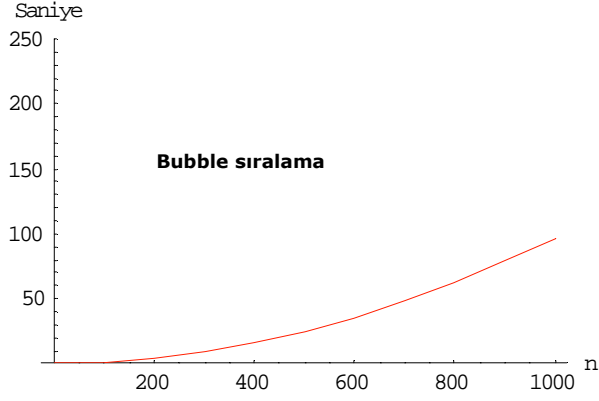
Not: Uygulamada genellikle

1. $a(n)$ vektörü ana bellekte değil, bir çevre depolama biriminde (hard disk, flash bellek) üzerindedir.
2. Bir vektör sıralanınca o vektöre bağlı birçok bilginin de beraber taşınması gerekir. Örneğin vatandaşlık numarasına göre kişiler sıralanacaksa, kişilerin adı, soyadı, v.s. gibi bilgilerinin de beraber taşınması veya bu bilgilerin adresinin de taşınması zorunludur.

Açıkçası sıralama problemi gerçekte daha karmaşıktır. Burada verilen alt programlar temel alınarak amaca cevap verecek şekilde iyileştirilebilirler.

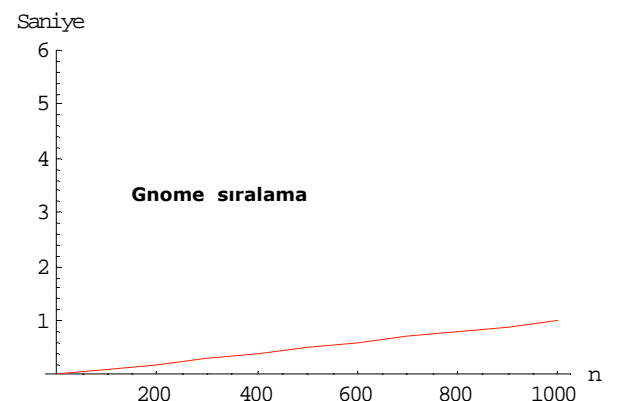
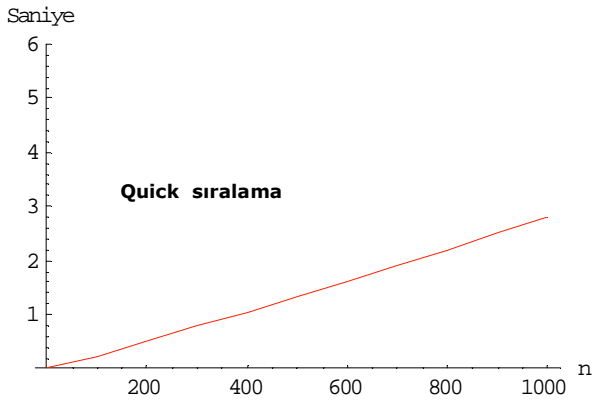
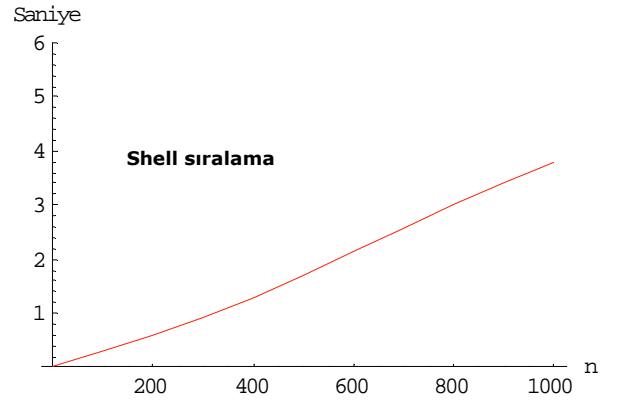
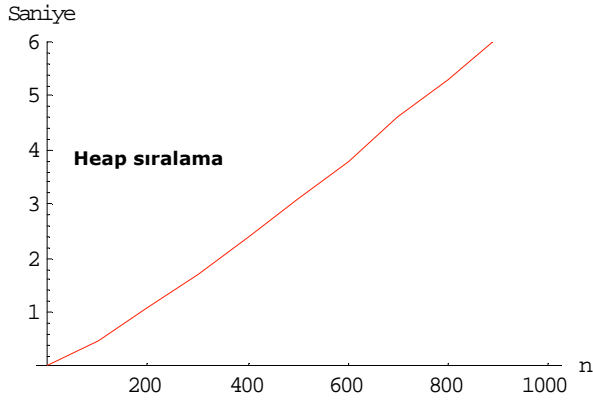
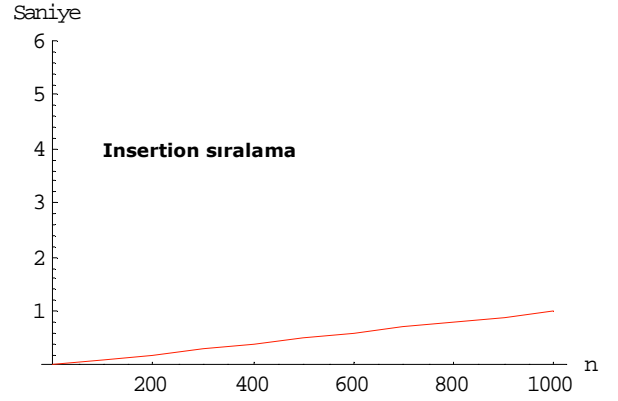
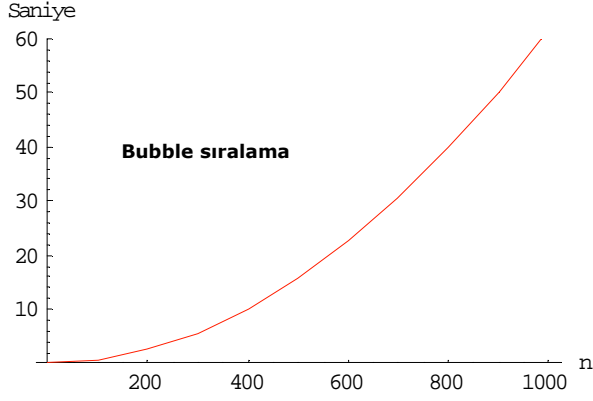
Karşılaştırma

t tipi: $a=[n, n-1, \dots, 2, 1]$ tamamen ters sıralı sayısal vektör için:



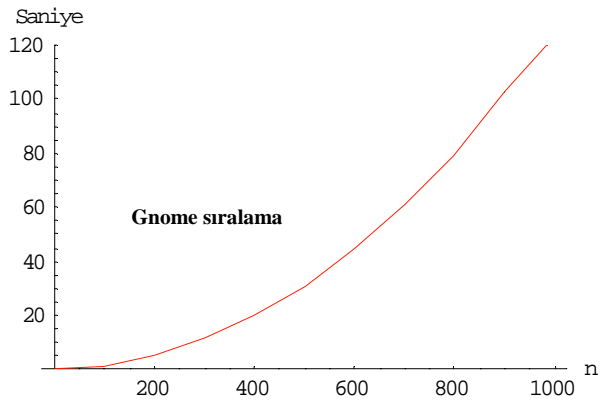
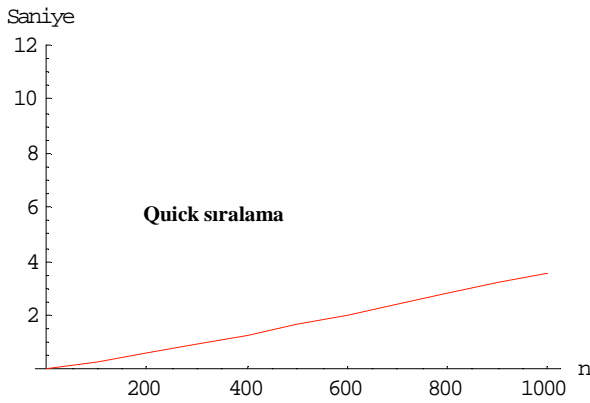
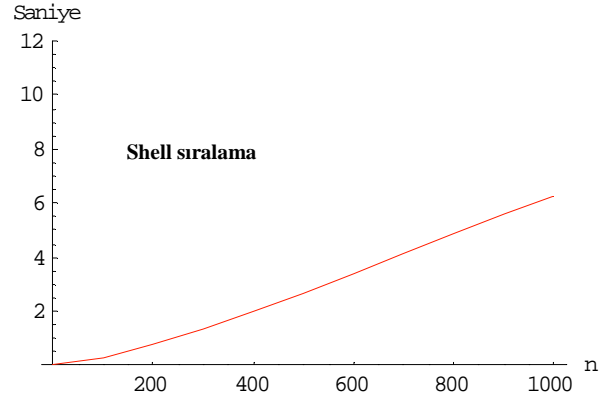
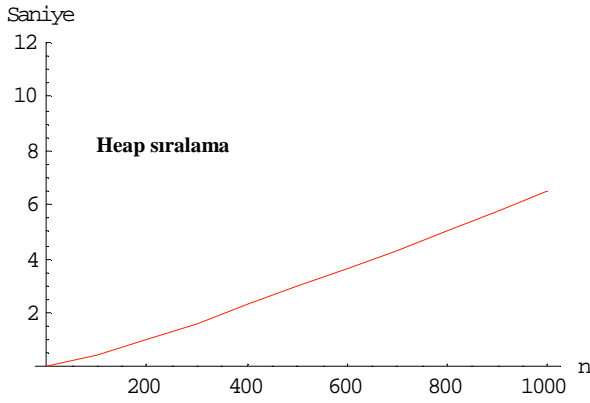
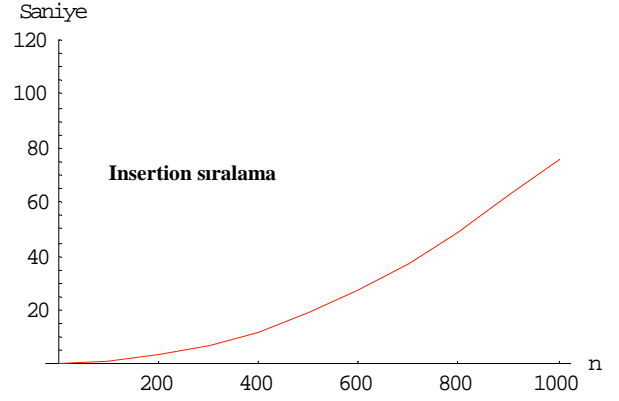
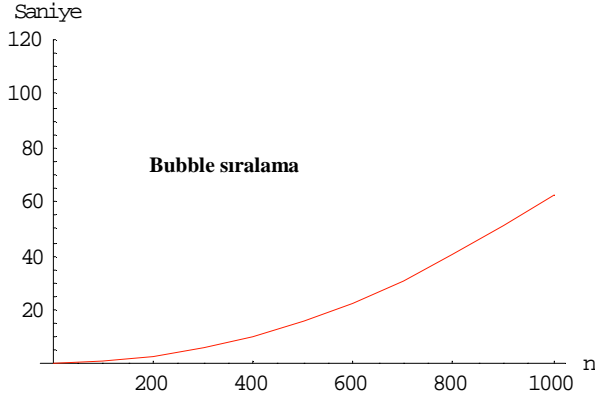
Tamamen ters sıralı vektör için Gnome en yavaş, Quick en hızlı sıralamadır.

h tipi: $a=[2,3, \dots, n-1, n, 1]$ hemen hemen sıralı sayısal vektör için:



Hemen hemen sıralı vektör için Bubble en yavaş, Insertion ve Gnome en hızlı sıralamadır.

r tipi: $\underline{a}=[\dots$ rasgele sayılar ...] rasgele sayılardan oluşan sırasız vektör için:



Rasgele sayılardan oluşan sırasız bir vektör için Gnome en yavaş, Quick en hızlı sıralamadır.

SONUÇ: En iyi sıralama metodu hangisidir?

1. Sıralanacak vektör hemen hemen sıralı ise: Uygulamada çoğu kez vektör sıralıdır ve zaman zaman yeni bir eleman eklenmesi ve yeniden sıralanması gerekir. Bu durumda en hızlı sıralama metodu Insertion veya Gnome sıralamadır, Insertion öncelikle tercih edilir.
2. Diğer her tip vektör için: Quick sıralama.

```

'-----Sıralama(Sorting) test-----
' Ahmet TOPÇU, Anadolu Üniversitesi, Eskişehir,1985
' Eskişehir Osmangazi Üniversitesi, 2010

' Sayısal vektörlerin sıralama yöntemleri:
' Bubble Sort, Insertion Sort, Heap Sort, Shell Sort
' QuickSort, GnomeSort

' a(n):sıralanacak vektör
' b(n):yardımcı vektör(sadece test için gerekli)
' n:vektörün eleman sayısı
'-----

DEFINT I-N
DEFSNG A-H, O-Z
DECLARE SUB BubbleSort (n, a())
DECLARE SUB insertionSort (n, a())
DECLARE SUB HeapSort (n, a())
DECLARE SUB ShellSort (n, a())
DECLARE SUB QuickSort (n, a())
DECLARE SUB GnomeSort (n, a())
DECLARE SUB VektorKur (n, b(), ihata)
DECLARE SUB Checkit (n, a())

DATA A Vektörünü kur,Bubble,Insertion,Heap,Shell,Quick,Gnome,Son
n = 100 ' a vektörünün eleman sayısı. test için n<=10000)
DIM s$(7), a(n), b(n)

Kuruldu = 0
FOR i = 0 TO 7
  READ s$(i)
NEXT i

100 CLS
FOR i = 0 TO 7
  PRINT i; s$(i)
NEXT i
IF Kuruldu = 0 THEN
  PRINT "Vektör kurulu değil!"
ELSE
' kurulan b yi a ya kopyala
  FOR i = 1 TO n
    a(i) = b(i): PRINT a(i);
  NEXT i
  PRINT : PRINT
  PRINT "n="; n; " Elemanlı vektör kurulu:"
  END IF

  PRINT
  INPUT "Seçenek="; iSecenek
  IF iSecenek = 7 THEN END
  IF iSecenek <> 0 AND Kuruldu = 0 THEN
    PRINT "önce vektörü kurunuz!"
    PRINT
    INPUT "Enter tuşuna basınız"; z$
    GOTO 100
  END IF

  t = TIMER
  SELECT CASE iSecenek
  CASE 0
    CALL VektorKur(n, b(), ihata)
    IF ihata <> 0 THEN GOTO 100
' kurulan b yi a ya kopyala
  FOR i = 1 TO n
    a(i) = b(i)
  NEXT i

  Kuruldu = 1
  CASE 1
    CALL BubbleSort(n, a())
  CASE 2
    CALL insertionSort(n, a())
  CASE 3
    CALL HeapSort(n, a())
  CASE 4
    CALL ShellSort(n, a())
  CASE 5
    CALL QuickSort(n, a())
  CASE 6
    CALL GnomeSort(n, a())
  CASE 7
    END

  CASE ELSE
    GOTO 100
  END SELECT

```

Sıralama ana programı

VektorKur alt programı

```

SUB VektorKur (n, b(), ihata)
'-----
' n elemanlı b(n) vektörünü sayılar ile doldurur

' t:ters sıralı sayılardan oluşan vektörü kur
' h:hemen hemen sıralı sayılardan oluşan vektörü kur
' r:rasgele sayılardan oluşan vektörü kur

' bu alt program sadece test için gereklidir.
'-----

' b vektörünün kurulması
ihata = 0
PRINT n; " elemanlı vektör kurulacak, vektör tipini seçin:"
INPUT "Vektörün tipi(t,h,r)="; Tip$

' Hemen hemen sıralı
IF Tip$ = "h" THEN
  FOR i = 1 TO n - 1
    b(i) = i + 1
  NEXT i
  b(n) = 1
  EXIT SUB
END IF

' Tamamen ters sıralı
IF Tip$ = "t" THEN
  FOR i = 1 TO n
    b(i) = n - i + 1
  NEXT i
  EXIT SUB
END IF

' rasgele
IF Tip$ = "r" THEN
  FOR i = 1 TO n
    b(i) = RND(1)
  NEXT i
ELSE
  ihata = 1
  PRINT "Yanlış veri=": INPUT "Enter tuşuna basınız"; z$
  END IF

END SUB 'VektorKur sonu

```

Sıralama ana programının devamı sonraki sayfada

```

Sure = TIMER - t
PRINT : PRINT s$(iSecenek); " n="; n; : PRINT

' sıralama doğru mu? Kontrol et
IF iSecenek <> 0 THEN CALL Checkit(n, a())
' a yı yazdır
FOR i = 1 TO n
  PRINT a(i);
NEXT i

PRINT : PRINT : PRINT s$(iSecenek); " n="; n; "Süre="; Sure; "Saniye"
INPUT "Enter tuşuna basınız"; z$
GOTO 100

END ' sıralama Test sonu

```

Sıralama ana programının devamı

```

SUB BubbleSort (n, a())
'-----
' gerçek sayılardan oluşan a(n) vektörü
' Bubble sıralama yöntemi ile küçükten
' büyüğe doğru sıralanır.
' n: vektörün eleman sayısı
'-----
FOR i = 1 TO n - 1
  k = i
  c = a(k)
  FOR j = i + 1 TO n
    IF a(j) < c THEN
      k = j
      c = a(k)
    END IF
  NEXT j
  b = a(i)
  a(i) = c
  a(k) = b
NEXT i

END SUB ' BubbleSort sonu

```

BubbleSort alt programı

```

SUB insertionSort (n, a())
'-----
' gerçek sayılardan oluşan a(n) vektörü
' Insertion sıralama yöntemi ile küçükten
' büyüğe doğru sıralanır.
' n: vektörün eleman sayısı
'-----
FOR j = 2 TO n
  i = j
  b = a(j)
40  i = i - 1
  IF a(i) > b THEN
    a(i + 1) = a(i)
    IF i > 0 GOTO 40
  END IF
  a(i + 1) = b
NEXT j

END SUB '

```

insertSort alt programı

```

SUB ShellSort (n, a())
'-----
' gerçek sayılardan oluşan a(n) vektörü
' Shell sıralama yöntemi ile küçükten
' büyüğe doğru sıralanır.
' n: vektörün eleman sayısı
'-----
L = 2 ^ INT(LOG(n) / LOG(2)) - 1
WHILE L > 2
  L = INT(L / 2)

  FOR j = 1 TO L
    FOR k = j + L TO n STEP L
      i = k
      b = a(i)

      DO
        IF a(i - L) <= b THEN EXIT DO
        a(i) = a(i - L)
        i = i - L
      LOOP WHILE i > L

      a(i) = b
    NEXT k
  NEXT j
WEND

END SUB 'ShellSort sonu

```

ShellSort alt programı

```

SUB GnomeSort (n, a())
'-----
' gerçek sayılardan oluşan a(n) vektörü
' Genome sıralama yöntemi ile küçükten
' büyüğe doğru sıralanır.
' n: vektörün eleman sayısı
'-----
i = 2
j = 2
WHILE i <= n
  IF a(i - 1) <= a(i) THEN
    i = j
    j = j + 1
  ELSE
    SWAP a(i - 1), a(i)
    i = i - 1
  END IF
  IF i = 1 THEN
    i = j
    j = j + 1
  END IF
END IF
WEND
EXIT SUB

END SUB ' GnomeSort sonu

```

GnomeSort alt programı

```

SUB HeapSort (n, a())
'-----
' gerçek sayılardan oluşan a(n) vektörü
' Heap sıralama yöntemi ile küçükten
' büyüğe doğru sıralanır.
' n: vektörün eleman sayısı
'-----

L = n / 2 + 1
m = n

1 IF L > 1 THEN
L = L - 1
b = a(L)
ELSE
b = a(m)
a(m) = a(1)
m = m - 1
IF m = 1 THEN
a(1) = b
EXIT SUB
END IF
END IF

i = L
j = L + L
2 IF j <= m THEN
IF j < m THEN
IF a(j) < a(j + 1) THEN j = j + 1
END IF
IF b < a(j) THEN
a(i) = a(j)
i = j
j = j + j
ELSE
j = m + 1
END IF
GOTO 2
END IF
a(i) = b

GOTO 1

END SUB 'HeapSort sonu

```

HeapSort alt programı

```

SUB QuickSort (n, a())
'-----
' gerçek sayılardan oluşan a(n) vektörü
' Quick sıralama yöntemi ile küçükten
' büyüğe doğru sıralanır.
' n: vektörün eleman sayısı
'-----

DIM L(40), iR(40)
iS1 = 1
L(1) = 1
iR(1) = n

WHILE iS1 > 0
L1 = L(iS1)
iR1 = iR(iS1)
iS1 = iS1 - 1

WHILE L1 < iR1
iL2 = L1
iR2 = iR1
b = a(INT((L1 + iR1) / 2))

WHILE iL2 <= iR2
WHILE a(iL2) < b
iL2 = iL2 + 1
WEND

WHILE b < a(iR2)
iR2 = iR2 - 1
WEND

IF iL2 <= iR2 THEN
SWAP a(iL2), a(iR2)
iL2 = iL2 + 1
iR2 = iR2 - 1
END IF
WEND

IF iL2 < iR1 THEN
iS1 = iS1 + 1
L(iS1) = iL2
iR(iS1) = iR1
END IF
iR1 = iR2
WEND
WEND

END SUB 'QuickSort sonu

```

QuickSort alt programı

```

SUB Checkit (n, a())
'-----
' sıralamanın doğru olup olmadığını kontrol eder.
' bu alt program sadece test için gereklidir
'-----

FOR i = 1 TO n - 1
IF a(i) > a(i + 1) THEN
PRINT : PRINT "Sıralama hatalı!";
INPUT "Enter tuşuna basınız!"; z$
EXIT SUB
END IF
NEXT i

END SUB ' Check it sonu

```

Checkit alt programı


```

'-----Ana program QuickAlfa test-----
' Ahmet TOPÇU, Anadolu Üniversitesi, 1985
' Satırları Türkçe harf dizini içeren a$(n) vektörü
' Quick sıralama yöntemi ile A dan Z ye alfabetik sıralanır
' n: vektörün satır sayısı
' iEsas: Sıralamaya esas alınacak karakter sayısı
' iEsas=0: dizinin tüm karakterlerini dikkate al ve sırala
' iEsas=1 Dizinin sadece ilk karakterini dikkate al ve sırala
' iEsas=3 Dizinin sadece ilk üç karakterini dikkate al ve sırala
'-----

DATA 40 : ' n nin değeri
DATA 0 : ' iEsas değeri(satırdaki tüm karakterleri dikkate al)

' A$ in satır verileri
DATA İbrahim,Şükrü,Ünal,Defne TOPÇU,Şener ZOR
DATA Şebnem AY,Çetin ÇINAR,Ömer BAĞ,Ali ÇAKIR,Sema
DATA Uğur,Oğuz,Şengül,Çakır-2,Ümit Kara
DATA İlker,Emre TAÇ,Şirin,Adnan,Betül YAŞ
DATA Filiz,Cem,Gülşen,Havva TOPÇU,Çakır-1
DATA Ziya,Yusuf TOPÇU,Işık,Tamer,Yelda
DATA Çağrı Şen,Çağatay,Celen,Özer,Kemal ER
DATA Timur,Özcan,İnci,Irmak,Şebnem TAŞ

DEFINT I-N

DECLARE SUB QuickSortAlfa (n, a$( ))
DECLARE SUB TurkceKodaDondur (n, a$( ), iEsas, TurkceKod$)
DECLARE SUB TurkceKodTablosu (TR$, TurkceKod$)
DECLARE SUB TurkceKaraktereDondur (n, a$( ), iEsas, TR$)

CLS
READ n ' a$ in satır sayısı
DIM a$(n) ' a$ için yer ayır

READ iEsas ' sıralamaya esas alınacak karakter sayısı
FOR i = 1 TO n
  READ a$(i)
NEXT i

PRINT "Sıralanmamış vektör (QuickAlfa):"
FOR i = 1 TO n
  PRINT a$(i)
NEXT i
PRINT
INPUT "Enter tuşuna basınız"; b$

CALL TurkceKodTablosu(TR$, TurkceKod$)
CALL TurkceKodaDondur(n, a$( ), iEsas, TurkceKod$)

CALL QuickSortAlfa(n, a$( ))

CALL TurkceKaraktereDondur(n, a$( ), iEsas, TR$)

CLS
PRINT "Sıralanmış vektör (QuickAlfa):"
FOR i = 1 TO n
  PRINT a$(i)
NEXT i

END ' Quick Alfa Test sonu

```

QuickAlfa ana programı

Buradaki Türkçe karakterler program Windows ortamına taşındığında düzgün görüntülenmez, taşındıktan sonra düzeltilmelidir

TurkceKodTablosu alt programı

```

SUB TurkceKodTablosu (TR$, TurkceKod$)
'-----
' Türkçe karakter tablosu ve kodlarını hazırlar
'-----
' Türkçe harfler ve rakamlar:
TR$ = "AaBbCcÇçDdEeFfGgĞğHhİiİjKkLlMmNnOoÖöPpQqRrSsŞşTtUuÜüVvWwXxYyZz"
TR$ = TR$ + "0123456789"
TR$ = TR$ + SPACE$(255 - LEN(TR$))

' TR$ içindeki her karaktere 1,2,3,.. kodlarını ata
TurkceKod$ = SPACE$(255) 'TurkceKod->türkçe karşılığı vektörü
FOR i = 1 TO 255
  k = ASC(MID$(TR$, i, 1))
  MID$(TurkceKod$, k, 1) = CHR$(i)
NEXT i

END SUB 'TurkceKodTablosu sonu

```

Buradaki Türkçe karakterler program Windows ortamına taşındığında düzgün görüntülenmez, taşındıktan sonra düzeltilmelidir

```

SUB TurkceKaraktereDondur (n, a$( ), iEsas, TR$)
'-----
' a$(n) in her satırının ilk iEsas karakterini
' Türkçe karaktere geri döndürür
'-----
FOR i = 1 TO n
  L = iEsas
  L1 = LEN(a$(i))
  IF iEsas >= 1 AND iEsas < L1 THEN
    L = iEsasa
  ELSE
    L = L1
  END IF

  FOR j = 1 TO L
    k = ASC(MID$(a$(i), j, 1))
    MID$(a$(i), j, 1) = MID$(TR$, k, 1)
  NEXT j
NEXT i

END SUB ' TurkceKaraktereDondur sonu

SUB TurkceKodaDondur (n, a$( ), iEsas, TurkceKod$)
'-----
' A$(n) in her satırının ilk iEsas karakterinin
' kodunu program koduna çevirir
' iEsas: Sıralamaya esas alınacak karakter sayısı
' iEsas=0: dizinin tüm karakterlerini dikkate al ve sırala
' iEsas=1: Dizinin sadece ilk karakterini dikkate al ve sırala
' iEsas=3: Dizinin sadece ilk üç karakterini dikkate al ve sırala
'-----
FOR i = 1 TO n
  L = iEsas
  L1 = LEN(a$(i))
  IF iEsas >= 1 AND iEsas < L1 THEN
    L = iEsasa
  ELSE
    L = L1
  END IF

  FOR j = 1 TO L
    k = ASC(MID$(a$(i), j, 1))
    MID$(a$(i), j, 1) = MID$(TurkceKod$, k, 1)
  NEXT j
NEXT i

END SUB ' TurkceKodaDondur sonu

```

TurkceKaraktereDondur
TurkceKodaDondur
Alt programları

QuickAlfa alt programı

```

SUB QuickSortAlfa (n, a$( ))
'-----
' Ahmet TOPÇU, Anadolu Üniversitesi, 1985
' Satırları Türkçe harf dizini içeren a$(n) vektörü
' Quick sıralama yöntemi ile A dan Z ye alfabetik sıralanır
' büyüğe doğru sıralanır.
' n: vektörün satır sayısı
'-----
DIM L(40), iR(40)
' index1-index2 arasını sırala
index1 = 1
index2 = n
IF index1 < 1 OR index1 > index2 THEN EXIT SUB
IF index2 < index1 OR index2 > n THEN EXIT SUB

index1 = 1
L(1) = 1
iR(1) = index2

WHILE index1 > 0
  L1 = L(index1)
  iR1 = iR(index1)
  index1 = index1 - 1

  WHILE L1 < iR1
    iL2 = L1
    iR2 = iR1
    b$ = a$(INT((L1 + iR1) / 2))

    WHILE iL2 <= iR2
      WHILE a$(iL2) < b$
        iL2 = iL2 + 1
      WEND

      WHILE b$ < a$(iR2)
        iR2 = iR2 - 1
      WEND

      IF iL2 <= iR2 THEN
        SWAP a$(iL2), a$(iR2)
        iL2 = iL2 + 1
        iR2 = iR2 - 1
      END IF
    WEND

    IF iL2 < iR1 THEN
      index1 = index1 + 1
      L(index1) = iL2
      iR(index1) = iR1
    END IF
    iR1 = iR2
  WEND
WEND

END SUB ' QuickAlfa sonu

```