



11. HAFTA

BLM102

PROGRAMLAMA DİLLERİ II

Yrd. Doç. Dr. Baha ŞEN

baha.sen@karabuk.edu.tr

KBUZEM

Karabük Üniversitesi

Uzaktan Eğitim Araştırma ve Uygulama Merkezi

1.3. Kabarcık Sıralama (Bubble Sort) Algoritması

Yer deęiřtirme sıralama (Exchange sort) olarak da bilinen anlařılması kolay bir algoritmadır. Dizide yer alan her eleman, sırayla kendisinden sonra gelen elemanla karřılařtırılır ve gerekiyorsa yer deęiřtirilir. Sıralama iřlemi yer deęiřtirme olduęu sũrece devam eder.

Kabarcık sıralamada dizi elemanlarının sayısının fazla olması durumunda iřlem zamanı artar.

Kabarcık algoritma kullanarak dizi elemanlarının artan řekilde sıralanması iin gerekli iřlem basamakları sıralanırsa;

1. Sıraya konulmamıř elemanların her birinin deęeri, bir sonraki deęerle yani komřu ile karřılařtırılır.
2. Eęer karřılařtırılan deęer komřu elemandan daha bũyũkse, komřusu ile yer deęiřtirilir. Bũylece sıralı olmayan elemanlar her tarandıęında, sadece yan yana bulunan iki eleman arasında sıralama yapılmıř olur. Dizinin bařından sonuna kadar tũm elemanlar bir kez iřleme tabi tutulduęunda dizinin son elemanı en bũyũk eleman haline gelecektir. (Yani ilk geiře, dizi ierisindeki en bũyũk eleman e sona gider.)
3. Tũm elemanlar sıralanana kadar, sıralanmamıř elemanların taranma iřlemi tekrarlanır.

28		28		28	28		28	28		28	28		28	28		28
47		47		47	16		16	16		16	16		16	16		16
16	→	16	→	16	47	→	47	24	→	24	24	→	24	24	→	24
24		24		24	24		24	47		47	11		11	11		11
11		11		11	11		11	11		11	47		47	33		33
33		33		33	33		33	33		33	33		33	47		47

Örnek şema incelenirse iç döngünün bir kez işletilmesi sonucu, dizideki en büyük değer en alta gider. Daha sonra tekrar aynı tarama yapılır. Ama bu sefer en sona kadar değil 5. Elemana kadar gidilir. Bu şekilde tüm liste taranır. Eğer son taramada bir değişiklik olmamışsa tarama işlemi bitirilir. Sonuçta küçükten büyüğe sıralanmış bir dizi elde edilir. İlk taramada görüleceği gibi elemanlardan bir örnekte 47 en sona gitmiş ve sıralı hale gelmiştir. Her adımda dizi elemanlarının nasıl tarandığı aşağıdaki şemada gösterilmiştir.

**Sıralanacak
Dizi:**

28	47	16	24	11	33
----	----	----	----	----	----

**Başlangıç
Durumu:**

28	47	16	24	11	33
----	----	----	----	----	----

1. Adım

28	16	24	11	33	47
----	----	----	----	----	-----------

2. Adım

16	24	11	28	33	47
----	----	----	----	-----------	----

3. Adım

16	11	24	28	33	47
----	----	----	-----------	----	----

4. Adım

11	16	24	28	33	47
----	----	-----------	----	----	----

5. Adım

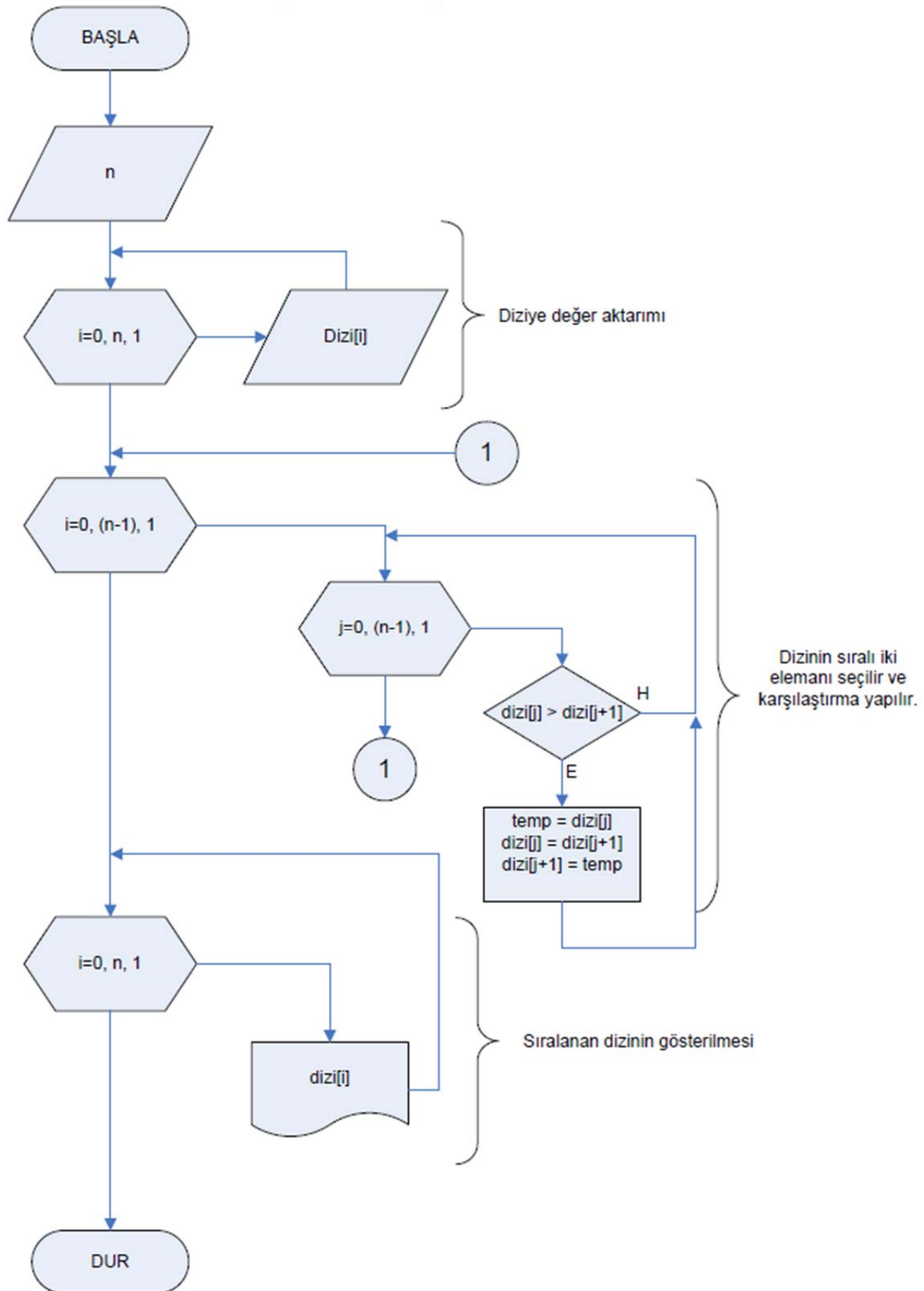
11	16	24	28	33	47
----	-----------	----	----	----	----

6. Adım

11	16	24	28	33	47
-----------	----	----	----	----	----

Akış Şeması

Kabarcık Sıralama (Bubble Sort)



1.4. Hızlı Sıralama (Quick Sort) Algoritması

Hızlı sıralama algoritması böl ve yönet politikasına dayanır; sıralanması istenen dizi belirli bir karşılaştırma (pivot) değerine göre iki alt diziye ayrılır. İkiye ayırma işlemi, bu karşılaştırma değerinden küçük olan elemanlar bir tarafta, büyük olan elemanlar diğer tarafta yerleştirilerek yapılır. Bu işlem geriye kalan her bölme üzerinde tekrarlanır. Karşılaştırma değeri ya rastgele seçilir ya da dizideki az sayıda elemanın değerinin ortalaması alınır. Hızlı sıralama algoritması en etkin (en hızlı çalışan) algoritma olmasına rağmen az sayıda elemana sahip dizilerde veya zaten neredeyse sıralı olan dizilerde tekrarlanan çağrılardan dolayı tercih edilmeyebilir. Hızlı arama algoritmasında recursive (özyinelemeli) yapı kullanılır.

Hızlı sıralama algoritmasını kullanarak dizi elemanlarını artan şekilde sıralamak için gerekli işlem basamakları:

1. İlk olarak sıralanacak diziyi ikiye bölmek için bir pivot (karşılaştırma değeri) seçilir. Pivot genellikle verilen dizinin ilk elemanı ya da son elemanı olabilir. Dizide pivottan büyük elemanlar pivotun sağına (üst), pivottan küçük elemanlar ise pivotun soluna (alt) konur. Pivot ise oluşan bu iki kümenin ortasına (orta) yerleşir. Böylece verilen dizi birbirinden bağımsız olarak iki alt diziye ayrılmış olur.
2. Hızlı sıralama algoritması bağımsız bu iki alt dizi (üst ve alt) içerisinde de **recursive** olarak çağrılır ve bu diziler kendi içerisinde 1.adım tekrarlanarak ikiye ayrılırlar. Bu işlemler diziler parçalanmayacak duruma gelene kadar tekrarlanır.

A Dizisi

Alt	Orta	Üst
Pivot elemanından küçük elemanlar	Pivot elemanı	Pivot elemanından büyük elemanlar

Örneğin;

Sıralanacak dizi;

14	5	83	23	4	87	13
----	---	----	----	---	----	----

Başlangıç
Durumu;

<u>14</u>	5	83	23	4	87	13
-----------	---	----	----	---	----	-----------

Pivot

1.Adım

4	5	83	23	14	87	13
----------	---	----	----	-----------	----	----

2.Adım

4	5	83	23	14	87	13
---	----------	-----------	----	----	----	----

3.Adım

4	5	13	23	14	87	83
---	---	-----------	----	----	----	-----------

4.Adım

Alt dizi sıralanır.	13	Üst dizi sıralanır.
---------------------	-----------	------------------------

5.Adım

4	5	13	14	23	83	87
---	---	----	----	----	----	----

Soldan 13'den büyük
ve sağdan 13'den
küçük sayılar ortada
buluştuğu veya
çakıştığı için pivotun
yeri bulunmuş olur
ve araya yerleştirilir.

1. Sıralanacak dizinin son sayısı (13) pivot elemanı (karşılaştırma) olarak seçilir. Bu eleman daha sonraki arama ve yer değiştirme işlemlerine tabi olmaz.
2. Sol başta pivot elemanı 13'ten büyük olan ilk sayı bulunur ve sağ baştan 13'den küçük ilk sayı bulunur. Daha sonra bu iki sayı yer değiştirilir.
3. Soldan pivot elemanından (13) büyük ve sağdan pivot elemanından küçük sayılar ortada buluşana kadar 2.adım tekrarlanır.

4. 3.adımda soldan 13' ten büyük ve sağdan 13'ten küçük sayılar ortada bulunduğu için (5 ve 83) 13 ile 83 yer değiştirir. Pivotun solundaki alt dizi ve sağındaki üst dizi kendi içinde aynı teknikle sıralanır.

Akış Şeması

