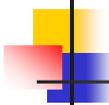




BM 210 Algoritma Analizi (Analysis of Algorithms)

Hazırlayan: M.Ali Akcayol
Gazi Üniversitesi
Bilgisayar Mühendisliği Bölümü



Konular

Comparison

- Selection Sort
- Insertion Sort
- Bubble Sort
- Shell Sort
- Merge Sort
- Quick Sort
- Heap Sort

No comparison

- Count Sort
- Radix Sort
- Bucket Sort

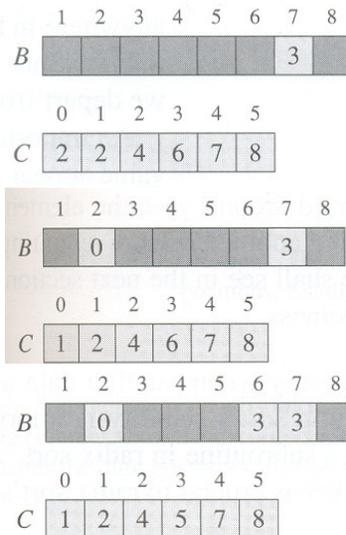
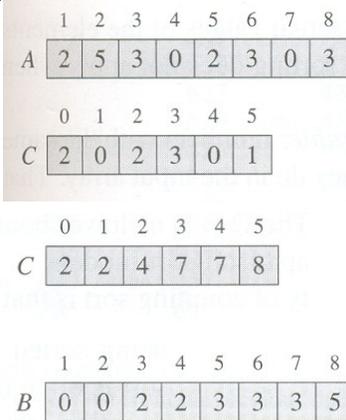


Count Sort

- n adet girişin integer olduğu kabul edilir
- Girişlerin 0 ile k arasında olduğu kabul edilir
- Temel olarak bir x elemanı için kendisinden küçük elemanların sayısını bulmayı amaçlar
- Örneğin x elemanından küçük 17 eleman varsa x elemanının doğru yeri 18 olur
- Girilen dizi boyutunda bir ek diziye ihtiyaç duyar
- Elemanların değer aralığı kadar elemana sahip ikinci bir ek diziye ihtiyaç duyar
- Stable sort özelliğine sahip algoritmadır



Count Sort





Count Sort

COUNTING-SORT(A, B, k)

$\Theta(k)$ 1 **for** $i \leftarrow 0$ **to** k
2 **do** $C[i] \leftarrow 0$

$\Theta(n)$ 3 **for** $j \leftarrow 1$ **to** $length[A]$
4 **do** $C[A[j]] \leftarrow C[A[j]] + 1$
5 ▷ $C[i]$ now contains the number of elements equal to i .

$\Theta(k)$ 6 **for** $i \leftarrow 1$ **to** k
7 **do** $C[i] \leftarrow C[i] + C[i - 1]$
8 ▷ $C[i]$ now contains the number of elements less than or equal to i .

$\Theta(n)$ 9 **for** $j \leftarrow length[A]$ **downto** 1
10 **do** $B[C[A[j]]] \leftarrow A[j]$
11 $C[A[j]] \leftarrow C[A[j]] - 1$

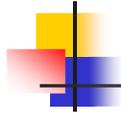
A, Sıralanacak dizi
B, Sıralanmış dizi
k, Değer aralığı

$\Theta(k+n)$ Genellikle $k = O(n)$ olur. Toplam çalışma süresi doğrusaldır ($\Theta(n)$)



Radix Sort

- En az öneme sahip basamaktan başlayarak sıralama işlemi yapar
- Her basamağın sıralamasında stable özelliğine sahip bir algoritma kullanılır
- Birden fazla anahtara göre sıralama gerektiğinde kolaylıkla kullanılabilir (Örn. tarihe göre sıralamada yıl, ay ve gün'e göre sıralama yapılır)
- Tarih sıralaması radix sort ile önce gün, sonra ay ve yıl'a göre stable sort yapılarak kolayca sıralanır



Radix Sort

329	720	720	329
457	355	329	355
657	436	436	436
839	457	839	457
436	657	355	657
720	329	457	720
355	839	657	839



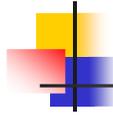
Radix Sort

RADIX-SORT(A, d)

$\Theta(d)$ 1 for $i \leftarrow 1$ to d
 $\Theta(n+k)$ 2 do use a stable sort to sort array A on digit i

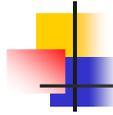
$\Theta(d(n+k))$

d (digit) sabit bir sayı ve $k = O(n)$ olduğundan toplam çalışma süresi doğrusaldır ($\Theta(n)$)



Radix Sort

now	sob	cab	ace	rap	tip	gig	joy
for	nob	wad	ago	gig	rap	dim	men
tip	cab	tag	and	wee	tap	tip	nob
ilk	wad	jam	bet	was	for	sky	now
dim	and	rap	cab	cab	tar	ilk	owl
tag	ace	tap	caw	wad	was	and	rap
jot	wee	tar	cue	tap	jot	sob	raw
sob	cue	was	dim	caw	hut	nob	sky
nob	fee	caw	dug	cue	bet	for	sob
sky	tag	raw	egg	fee	you	jot	tag
hut	egg	jay	fee	raw	now	you	tap
ace	gig	ace	few	ago	few	now	tar
bet	dug	wee	for	tar	caw	joy	tip
men	ilk	fee	gig	jam	raw	cue	wad
egg	owl	men	hut	dug	sky	dug	was
few	dim	bet	ilk	you	jay	hut	wee
jay	jam	few	jam	and	joy	owl	you
owl	men	egg	jay				
joy	ago	ago	jot				



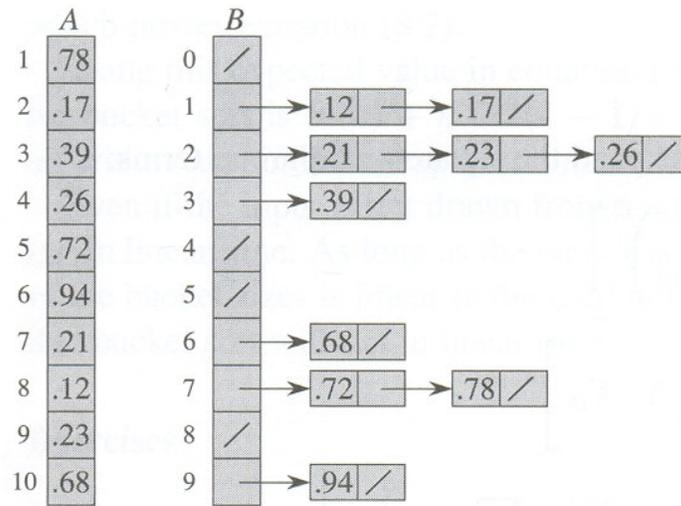
Radix Sort

now	ace	ace	ace	joy	jot	jot	jot
for	ago	ago	ago	rap	joy	joy	joy
tip	and	and	and	gig	men	men	men
ilk	bet	bet	bet	wee	now	now	nob
dim	cab	cab	cab	was	nob	nob	now
tag	caw	caw	caw	cab	owl	owl	owl
jot	cue	cue	cue	wad	rap	rap	rap
sob	dim	dim	dim	caw	sob	sky	sky
nob	dug	dug	dug	cue	sky	sob	sob
sky	egg	egg	egg	fee	tip	tag	tag
hut	for	few	fee	tap	tag	tap	tap
ace	fee	fee	few	ago	tap	tar	tar
bet	few	for	for	tar	tar	tip	tip
men	gig	gig	gig	jam	wee	wad	wad
egg	hut	hut	hut	dug	was	was	was
few	ilk	ilk	ilk	and	wad	wee	wee
jay	jam	jay	jam				
owl	jay	jam	jay				

Bucket Sort

- Bucket sort doğrusal zamanda çalışır
- Girişin düzgün dağılımlı olduğu kabul edilir
- Random olarak $[0, 1)$ aralığında oluşturulmuş giriş bilgileri olduğu kabul edilir
- Temel olarak $[0, 1)$ aralığını n adet eşit alt aralığa böler
- Girişi bu aralıklara dağıtır
- Aralıklardaki değerleri insertion sort ile sıralar
- Aralıkları birbiri ardına ekleyerek sıralanmış diziyi elde eder

Bucket Sort





Bucket Sort

BUCKET-SORT(A)

$\Theta(n)$ 1 $n \leftarrow \text{length}[A]$
 $\Theta(n)$ 2 **for** $i \leftarrow 1$ **to** n
 $\Theta(n)$ 3 **do** insert $A[i]$ into list $B[\lfloor nA[i] \rfloor]$
 $\Theta(n)$ 4 **for** $i \leftarrow 0$ **to** $n - 1$
 $O(n_i^2)$ 5 **do** sort list $B[i]$ with insertion sort
 $\Theta(n)$ 6 concatenate the lists $B[0], B[1], \dots, B[n - 1]$ together in order

$\Theta(n)$ $O(n_i^2)$ $B[i]$. yerdeki eleman sayısını ifade eder. Beklenen çalışma süresi doğrusaldır ($\Theta(n)$)



Haftalık Ödev

- Count sort, Radix sort ve Bucket sort algoritmaları için bir program yazınız. Programda seçilen algoritmanın çalışması ve girişlerdeki ekranda gösterilecektir. Program C# ile yazılacak ve görsel arabirime sahip olacaktır.

Not: Ödev programın çıktısıyla birlikte diskete kaydedilerek teslim edilecektir. Ödevler en geç 19 Nisan 2005 tarihine kadar verilecektir.