

Problem Seti 9 Çözümleri

Problem 9-1. Daha fazla paralel birleştirme sıralaması

Bu problemde, dersteki paralel birleştirme sıralaması algoritmasını geliştireceğiz. Sınıfta anlatılan algoritmanın işi $\Theta(n \lg n)$, paralelliği ise $\Theta(n/\lg^2 n)$ düzeyindeydi. Burada aynı miktar işe karşılık daha fazla paralelliği olan bir algoritma geliştireceğiz.

- (a) Toplam n sayıda elemanı olan iki sıralı dizilimin, n elemanlarının ortancasını tek işlemci kullanarak $\Theta(\lg n)$ süresinde bulacak algoritmayı verin.

Çözüm: Temel fikir, size A ve B gibi iki dizilim verildiğinde, eğer her ikisinin de uzunluğunu biliyorsanız, bir $A[i]$ elemanının ortanca olup olmadığını, sabit zamanda kontrol edebileceğinizdir. Ortancanın $A[i]$ olduğunu farzedin. Dizilim sıralı olduğundan; ortanca, A dizilimindeki değerlerin tam olarak $i - 1$ 'inden daha büyüktür. Ayrıca eğer ortanca ise, aynı zamanda, B 'deki $j = \lceil n/2 \rceil - (i - 1)$ elemandan da büyüktür. $B[j] \leq A[i] \leq B[j + 1]$ ifadesini kontrol etmek sabit zaman gerektirir. Eğer $A[i]$ ortanca değilse; $A[i]$ 'nin, $B[j]$ ve $B[j + 1]$ 'den büyük yada küçük olmasına bağlı olarak, $A[i]$ 'nin ortancadan büyük ya da küçük olduğunu bilirsiniz. Böylece, $A[i]$ 'nin, $\Theta(\lg n)$ (en kötü durum) süresinde ikili aramasını yapabilirsiniz. Sözde kodu ise şöyledir:

```
MEDIAN-SEARCH ( $A[1..l]$ ,  $B[1..m]$ ,  $left$ ,  $right$ )           (ortanca arama)
1 if  $left > right$ 
2   then return MEDIAN-SEARCH( $B$ ,  $A$ ,  $\max(1, \lceil n/2 \rceil - l)$ ,  $\min(m, \lceil n/2 \rceil)$ )
3  $i \leftarrow \lfloor (left + right)/2 \rfloor$ 
4  $j \leftarrow \lceil n/2 \rceil - i$ 
5 if  $(j = 0 \vee A[i] > B[j])$  ve  $(j = m \vee A[i] < B[j + 1])$ 
6   then return ( $A$ ,  $i$ )                                ► median =  $A[i]$ 
7 elseif  $(j = 0 \vee A[i] > B[j])$  ve  $j \neq m$  ve  $A[i] > B[j + 1]$ 
8   then return MEDIAN-SEARCH( $A$ ,  $B$ ,  $left$ ,  $i - 1$ )        ► median <  $A[i]$ 
9   else return MEDIAN-SEARCH ( $A$ ,  $B$ ,  $i + 1$ ,  $right$ ) ► median >  $A[i]$ 
```

$left$ ve $right$ (sol ve sağ) için varsayılan değerler öyle olsunlar ki;
MEDIAN-SEARCH (A , B) çağrıldığında bu aşağıdaki ifadeye eşit olsun:

MEDIAN-SEARCH($A[1..l]$, $B[1..m]$, $\max(1, \lceil n/2 \rceil - m)$, $\min(l, \lceil n/2 \rceil)$)

MEDIAN-SEARCH (A , B) 'nin içindeki değişmez öyledir ki; ortanca her zaman ya $A[left..right]$ 'ın ya da B 'nin içindedir. Bu ilk çağrı için geçerlidir, çünkü A ve B sıralıdır; bu nedenle; ortanca tanımı gereği, $\max(1, \lceil n/2 \rceil - m)$ veya $\min(l, \lceil n/2 \rceil)$ içinde yer alır. Bu 8 ve 9. satırlardaki özyinelemeli çağrılar için de geçerlidir, çünkü algoritma,

ortancanın tanımı gereği, içinde ortanca olamayacak kısımları dizilimden çıkarır. İkinci satırdaki özyineleme de değişmezi korur, çünkü eğer $left > right$ ise ortanca, B 'nin içinde, yeni $left(sol)$ ve $right(sağ)$ değerlerinin arasında olmalıdır. Algoritma sona erdiğinde, döndürülen değer tanım gereği ortancadır. Bu algoritmanın son bulacağı garantilidir, çünkü her adımda $sağ-sol$ azalmaktadır ve ortanca ya A 'da ya da B 'de olmalıdır. MEDIAN-SEARCH 'ün en kötü durum asimptotik koşma süresi, 2 ikili-arama ile aynıdır; yani $\Theta(\lg n)$ 'dir.

Ortancayı bulmanın başka bir yolu, iki dizilimin ortancalarını sürekli olarak kıyaslayıp, ortancayı içermeyecek kısımları atmaktır. Bu, en çok $2[\lg n]$ döngü süresince koşar, çünkü her döngü, girdi dizilimlerinden birinin yarısını atar. Bu nedenle bu algoritmanın da koşma süresi $\Theta(\lg n)$ 'dir.

- (b) Şık (a)'daki algoritmayı bir altprogram olarak kullanıp, iki sıralı dizilimi birleştirecek, bir çok izlekli algoritmayı verin. Algoritmanızın yaptığı iş $\Theta(n)$, ve paralelliği (yani dönemdeşliği) $\Theta(n/\lg^2 n)$ olsun. İş ve kritik-yol uzunluğu için gerekli yinelemeleri belirtin ve çözün; paralelliğin istendiği gibi $\Theta(n/\lg^2 n)$ olduğunu gösterin.

Çözüm:

Birleştirme amacıyla, sıralı dizilimlerin küçük ve büyük yarılarını, özyinelemeli ve paralel olarak birleştiren çok izlekli bir algoritma yazabiliriz:

```

P-MERGE (A[1 .. l], B[1 .. m], T[1 .. n])
1  if n ≤ 0
2  then return
3  (A, k) ← MEDIAN-SEARCH (A, B)   ► wlog ortancanın A'da olduğunu farzedin
4  if n = 1
5  then T[1] ← A[1]
6  else spawn P-MERGE(A[1 .. k - 1], B[1.. ⌈n/2⌉ - k - 1], T[1.. ⌈n/2⌉ - 1])
7          spawn P-MERGE(A[k .. l], B[⌈n/2⌉ - k .. m], T[⌈n/2⌉ .. n])
8  sync

```

Bu algoritma, iki sıralı dizilimi bir T arabelleği içinde birleştirir. Genellikle uzaklaşmadan, ortancanın A'nın içinde olduğunu varsayabiliriz; eğer B'nin içindeyse, algoritmanın geri kalan kısmında, A ile B'yi değiştirebiliriz.

Bu algoritmanın işi $T_1(n) = 2T_1(n/2) + \Theta(\lg n) = \Theta(n)$ 'dir. Kritik yol uzunluğu $T_\infty(n) = T_\infty(n/2) + \Theta(\lg n) = \Theta(\lg^2 n)$ olur. Bu nedenle paralellik $T_1/T_\infty = \Theta(n/\lg^2 n)$ olur.

- (c) *İsteğe bağlı:* Şık (a)'daki algoritmayı rastgele bir sıra istatistiği bulmak amacıyla genelleyin. Bu algoritmayı kullanarak, $(n/\lg n)$ işi ve $\Theta(n/\lg n)$ paralelliği sağlayan bir birleştirme-sıralaması algoritmasını açıklayın.

Çözüm:

Önce rastgele sıra istatistiklerini bulan bir altprograma ihtiyacımız var. Aşağıdaki algoritma, iki sıralı dizilimin tüm elemanlarının k 'ninci sıra istatistiğini bulur.

```

ORDER-STATISTICS( $A[p..q]$ ,  $B[r..s]$ ,  $k$ )    (sıra-istatistikleri)
1  if  $q \leq p + 1$  or  $s \leq r + 1$  ► Bir dizilimde 3'den az eleman vardır
2    then  $k$ 'ninci sıra istatistiğini sabit sürede apaçık bul ve döndür.
3  if  $k \leq s + q - p - r + 2$  ►  $k$ 'ninci eleman ortancaya küçük-eşitse
4    then if  $A[\lceil \frac{p+q}{2} \rceil] < B[\lceil \frac{r+s}{2} \rceil]$ 
5      then return ORDER-STATISTICS( $A[p..q]$ ,  $B[r..\lceil \frac{r+s}{2} \rceil]$ ,  $k$ )
6      else return ORDER-STATISTICS( $A[p..\lceil \frac{p+q}{2} \rceil]$ ,  $B[r..s]$ ,  $k$ )
7  if  $k > s + q - p - r + 2$  ►  $k$ th element is greater than the median element
8    then if  $A[\lceil \frac{p+q}{2} \rceil] < B[\lceil \frac{r+s}{2} \rceil]$ 
9      then return ORDER-STATISTICS( $A[\lceil \frac{p+q}{2} \rceil..q]$ ,  $B[r..s]$ ,  $k - \lceil \frac{p+q}{2} \rceil + p - 1$ )
10   else return ORDER-STATISTICS( $A[p..q]$ ,  $B[\lceil \frac{r+s}{2} \rceil..s]$ ,  $k - \lceil \frac{r+s}{2} \rceil + r - 1$ )

```

Şimdi bu altprogramı iki dizilimi birleştirmek için kullanırız. Dizilimleri, 2 alt-dizilime bölmek yerine, $(k = \sqrt{n}, 2\sqrt{n}, \dots, (\sqrt{n}-1)\sqrt{n})$ için $\sqrt{n}-1$ eşit aralıklı sıra istatistikleri bulur ve $\sqrt{n}$ alt-dizilimlerini paralel olarak birleştiririz. Böylece birleştirme altprogramının kritik yolu $M_{\infty}(n) = M_{\infty}(\sqrt{n}) + \lg n$ olur ve bu yinelemenin çözümü de $M_{\infty}n = \Theta(\lg n)$ olur.

Birleştirme işi $M_1(n) = \sqrt{n}M_1(\sqrt{n}) + \Theta(\sqrt{n} \lg n)$ 'dir ve çözümü de $M_1(n) = \Theta(n)$ olur. Böylece, birleştirme-sıralamasının işi $\Theta(n)$ değerinde kalırken, kritik yol $T_{\infty}(n) = T_{\infty}(n/2) + \Theta(\lg n) = \Theta(\lg^2 n)$ değerine azalır.