Algoritma Analizi

Ders 8

Doç. Dr. Mehmet Dinçer Erbaş
Bolu Abant İzzet Baysal Üniversitesi
Mühendislik Fakültesi
Bilgisayar Mühendisliği Bölümü

Medyan ve sıra istatistikleri

- n elemana sahip bir dizinin i numaralı sıra istatistiği bu dizinin i numaralı en küçük elemanıdır (i. en küçük eleman).
  - Örneğin diziye ait minimum eleman birinci sıra istatistiğidir (i = 1).
  - Diziye ait en maksimum eleman n. sıra istatistiğidir (i = n).
- Dizinin medyan elemanı dizide “ortadaki“ elemandır.
  - Dizi tek sayıda elemana sahip ise medyan i = (n+1)/2 sıralı elemandır.
  - Dizide çift sayıda eleman olduğu durumda ise iki farklı medyan değeri olur. Bunlar
    - i = n/2 sıralı eleman, yani alt medyan.
    - i = n/2 + 1 sıralı eleman, yani üst medyan.
  - Öyleyse dizideki eleman sayısının çift veya tek olmasından bağımsız olarak medyan $i=\lfloor (n+1)/2 \rfloor$ ve $i=\lceil (n+1)/2 \rceil$ sıralı elemanlarda bulunur.
  - Bu bölümde medyan dediğimizde alt medyandan bahsedeceğiz.

Medyan ve sıra istatistikleri

- Bu bölümde n farklı sayı içeren bir kümenin i numaralı sıra istatistiğinin bulunması konusunu inceleyeceğiz.
  - Problemi basitleştirmek için elemanların birbirlerinden farklı olduğunu farzedeceğiz.
    - Dizide aynı elemanlar birden fazla kere bulunursa problem benzer şekilde çözülebilir.
- Bir dizinin i numaralı sıra istatistiği seçme problemi şu şekilde tanımlanabilir:
  - Girdi: n farklı elemandan oluşan A dizisi ve i sayısı ($1 \leq i \leq n$).
  - Çıktı: Dizideki tam olarak i – 1 elemandan büyük olan $x \in A$ elemanı
- Bu problemi şu ana kadar öğrendiğimiz sıralama algoritmaları ile çözebiliriz.
  - Bu yöntemle problem O(n lg n) sürede çözülebilir.

Minimum ve maksimum

- Bir dizideki minimum değer kaç karşılaştırma yapılarak bulunabilir?
  - Dizideki o ana kadar bulduğumuz minimum değeri hatırlayarak her elemanı karşılaştırırsak
    - n – 1 karşılatırma sonucu minimum değer bulunabilir.
  - Daha az sayıda karşılaştırma yaparak minimum değeri bulmamılz mümkün müdür?
    - Hayır. Dizideki minimum değer bulma işlemi sırasında minimum değer dışındaki bütün elemanlar en az bir kere karşılaştırılmalıdır.
- Aynı şekilde bir dizideki maksimum değer bulunabilir.

Minimum ve maksimum

```
MINIMUM(A)
1  min = A[1]
2  for i = 2 to A.length
3      if min > A[i]
4          min = A[i]
5  return min
```

Minimum ve maksimum

- Bazı durumlarda bir dizideki hem minimum hem maksimum değeri bulmamız gerekebilir.
  - Bu işlemi önceki algoritmayı kullanarak 2(n – 1) karşılaştırma yaparak gerçekleştirebiliriz.
    - O(n) süre alır ve bu çözüm optimumdur.
  - Ancak bu işlemi daha az karşılaştırma yaparak çözebiliriz.
    - Her yeni elemanı o anki maksimum ve minimum ile karşılaştırmak yerine yeni elemanları ikili olarak inceliyoruz.
      - Önce iki elemanı karşılaştırır sonra büyük olan ile maksimumu, küçük olan ile ise minimumu karşılaştırırız.
    - Böylece her yeni iki eleman için 2*2 = 4 karşılaştırma yerine 3 karşılaştırma yapmamız yeterli olur.

Minimum ve maksimum

- Bazı durumlarda bir dizideki hem minimum hem maksimum değeri bulmamız gerekebilir.
  - Dizide tek sayıda eleman var ise ilk elemanı başlangıçta minimum ve maksimum olarak seçebiliriz.
  - Çift sayıda eleman var ise ilk iki elemanı karşılaştırarak başlangıçta minimum ve maksimum seçebiliriz.
  - Bu durumda n tek sayı olduğunda toplam $3\lfloor n/2 \rfloor$ karşılaştırma yaparız.
  - n çift sayı olduğunda ise toplam $3n/2-2$ karşılaştırma yaparız.
  - Öyleyse toplam yaptığımız karşılaştırma sayısı en fazla $3\lfloor n/2 \rfloor$ olur.

Doğrusal zamanda beklenilen seçme

- Bir serideki i. sıra istatistiğini bulma problemi Θ(n) sürede çözülebilir.
- Bu bölümde bu amaçla kullanılabilecek bir algoritmayı tanıtacağız.
- Algoritma Quicksort algoritmasında gördüğümüz yöntem ile çalışıyor.
  - Hızlı sıralama algoritmasında olduğu gibi verilen girdiği iki parçaya ayırıyor.
  - Ancak her iki parça üzerinde yinelemeli olarak çalışmak yerine oluşan parçalardan sadece biri üzerinde çalışıyor.

Doğrusal zamanda beklenilen seçme

```
RANDOMIZED-SELECT(A, p, r, i)
1  if p == r
2       return A[p]
3  q = RANDOMIZED-PARTITION(A, p, r)
4  k = q − p + 1
5  if i == k              // the pivot value is the answer
6       return A[q]
7  elseif i < k
8       return RANDOMIZED-SELECT(A, p, q − 1, i)
9  else return RANDOMIZED-SELECT(A, q + 1, r, i − k)
```

Doğrusal zamanda beklenilen seçme

- Birinci satırda yinelemeli fonksiyonun başlangıç durumu kontrol ediliyor.
- Üçüncü satırda verilen dizi iki parçaya ayrılıyor.
- Dördüncü satırda önceki adımda seçilen pivot elemanın hangi pozisyona yerleştirildiği hesaplanıyor.
- Beşinci satırda yerleştirilen pozisyonun aranan sıra istatistiğine eşit olup olmadığı kontrol ediliyor.
- Eşit değil ise k sayısı ile aranan sıra istatistiği karşılatırılarak oluşturulan iki altdizinin hangisinin içerisinde elemanın aranması gerektiği bulunuyor.
- Buna göre belirtilen altdizi için fonksiyon yinelemeli olarak tekrar çağırılıyor.

Doğrusal zamanda beklenilen seçme

- En kötü durumda her sefer içerisinde aradığımız dizi büyüklüğü bir azalır.
  - Bu durumda algoritmanın çalışma süresi $\Theta(n^2)$.
- Algoritmada pivot olarak rastgele bir eleman seçiliyor.
  - Her sefer en büyük veya en küçük elemanın seçilme ihtimali son derece düşük.
- Bu sebeple algoritmanın çalışma süresi ortalamada en kötü zamandan daha hızlı.
- Girdi olarak verilen n elemanlı bir dizide ortalama beklenilen çalışma süresi, T(n), şu şekide hesaplanabilir.
  - Geri kalan kısmı tahtada açıklanacak.

En kötü durumda doğrusal zamanda seçme

- Bu bölümde en kötü durumda O(n) sürede çalışan bir seçme algoritmasını inceleyeceğiz.
- Bir önceki algoritmada olduğu gibi yinelemeli fonksiyonlar kullanılarak bir dizideki istenilen sıra istatistiğine sahip eleman bulunacak.
- Bu algoritmadaki ana fikir, parçalama sırasında birbirine yakın büyüklükte altdiziler oluşturmayı garanti etmektir.
- SELECT isimli algoritma aşağıda belirtilen adımlardan oluşur:
  - 1. Verilen n elemanlı diziyi $\lfloor n/5 \rfloor$ tane 5 elemanlı altdizi ve 5'e bölündüğünde kalan elemanlardan oluşan bir altdiziye böl.
  - 2. Insertion sort kullanarak her bir 5 elemanlı altdizinin medyanını bul ve bu medyan değerlerini al.
  - 3. SELECT fonksiyonunu yinelemeli olarak çağırarak önceki adımda bulunan medyan değerlerinin medyanını bul. Bu değere x diyelim.

En kötü durumda doğrusal zamanda seçme

- SELECT isimli algoritma aşağıda belirtilen adımlardan oluşur:
  - 4. x değerini kullanarak girdi olarak alınan diziyi parçala. k değeri parçalama sonrasında x değerinden düşük eleman sayısı olsun, yani x k numaralı en küçük eleman ve n – k tane x'den büyük eleman var.
  - 5. i = k ise, x değeri döner. Aksi halde SELECT fonksiyonunu yinelemeli olarak kullan ve i < k ise x değerinden düşük elemanlar için fonksiyonu çağır, i > k ise x değerinden yüksek elemanlar için fonksiyonu çağır.

En kötü durumda doğrusal zamanda seçim