

Algoritma Analizi

Ders 1: Ders tanıtımı

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

Algoritma Analizi

- Bu derste aşağıda belirtilen derslerde öğrendikleriniz kullanılacaktır:
 - C Programlama
 - Veri Yapıları
 - Ayrık Matematik
- Yukarıda belirtilen derslerden herhangi birini almayanlar (veya tam öğrenemeyenler) bu derste zorlanabilirler.
- Özellikle 2. sınıf öğrencilerine bu dersi bu dönemde almayı tavsiye etmiyorum.

Algoritma

- Algoritma: Belirli bir işi yapmak üzere dizayn edilmiş, belli sayıda adımları olan, belirli girdiler alan ve sonuç olarak belirli çıktılar oluşturan işlemler bütünüdür.
 - ~ Bir algoritmayı girdileri çıktılarına dönüştüren işlem adımları olarak düşünebiliriz.
 - ~ Probleme ait girdi-çıkı ilişkisi tanımlandıktan sonra bu dönüşümü sağlayan algoritma oluşturulabilir.
- Problem örneği: elimizde bulunan bir sayı kümesinin elemanlarını azalmayan şekilde sıralamak
 - ~ Bu problem literatürde sıralama (İng: sorting) problemi olarak adlandırılır.

Algoritma

- Sıralama problemi tanımı:
 - ~ Girdi: n elemanlı bir sayı dizisi $\{a_1, a_2, \dots, a_n\}$
 - ~ Çıktı: verilen sayı dizisinin yeni bir sıralaması $\{a'_1, a'_2, \dots, a'_n\}$,
şöyleki $a'_1 \leq a'_2 \leq \dots \leq a'_n$.
- Örneğin verilen sayı dizisi $\{39, 41, 59, 26, 41, 58\}$ ise algoritma çalıştığında oluşturulan çık $\{26, 39, 41, 41, 58, 59\}$ olması beklenmektedir.
- Sıralama, birçok problemin çözümü için gerekebilen bir ara işlemdir.
- Bu nedenle bilgisayar bilimi konusunda incelenen temel bir problem olmuştur.
- Yapılan araştırmalar sayesinde sıralama konusunda birçok farklı algoritmaya sahibiz.

Algoritma

- Sıralama problemi

~ Belirli bir problem için seçilen sıralama algoritmasının uygunluğu, problemin sahip olduğu özelliklere göre değişkenlik gösterir. Bu özelliklerden bazıları şunlardır:

- Sıralanacak nesne sayısı
- Nesnelerin başlangıçta bir miktar sıralı olma durumu
- Nesnelerin sahip olabileceği değerler üzerinde kısıtlamalar
- Algoritmanın çalıştırılacağı bilgisayarın mimarisi
- Kullanılacak depolama araçlarının tipi

Algoritma

- Doğru (İng: correct) algoritma: Verilen her bir girdi için doğru çıktı ile sonlanan algoritmadır.
 - ~ Doğru bir algoritma verilen problemi çözer.
- Hatalı (İng: incorrect) algoritma bazı girdiler için yanlış çıktı ile sonlanabilir veya sonlanmayabilir.
- Algoritmalar mümkün olduğunca verimli (İng: efficient) olmalıdır, yani verilen problemi mümkün olan en kısa zamanda ve en az miktarda kaynak kullanarak çözmelidir.
- Program: Belirli bir programlama dilinde bir algoritmanın gerçekleştirilmiş halidir.
- Veri yapısı: Belli bir algoritmanın çalışabilmesi için gereken verilerin organize edilmesidir.

Algoritma

- Oldukça karmaşık algoritmaların geliştirildiği problemlere örnekler:
 - ~ İnsan gen haritası projesi (Human Genome Project)
 - ~ İnternet
 - ~ E-ticaret
 - ~ Üretim ve diğer ticari faaliyetler
 - ~ Harita üzerinde verilen noktaları en kısa yolu kullanarak ziyaret etme
 - ~ İki farklı sembol serisi verilmiş. Bunların sahip olduğu ortak en uzun altdiziyi (subsequence) bulma.
 - ~ Mekanik parçalardan oluşan bir kütüphane oluşturmak istiyoruz ve her bir parça diğerlerinin birleşmesinden oluşuyor. Bu parçaları öyle bir şekilde sıralamak istiyoruz ki her parça kendisini kullanan parçalardan önce sıralanacak.

Algoritma

- Problem tipleri ve geliştirilebilecek çözüm yöntemleri oldukça çeşitli. Ancak genel olarak şu özelliklere sahipler:
 - ~ Denenebilecek çözüm sayısı oldukça fazla ve bu çözümlerin önemli bir miktarı aslında problemi çözmiyor. Bu sebeple en iyi çözümü bulmak oldukça zor.
 - ~ Belirtilen problemlere çözüm bulunursa, bu çözüm yöntemlerinin uygulanabileceği uygulama alanları mevcut.
- Algoritmalar bilgisayar sistemleri üzerinde çalışırlar.
 - ~ Bu sistemler sonsuz hıza ve sonsuz kaynaklara sahip olsaydı, doğru çalışan herhangi bir algoritmayı uygulamak her problemi çözmek için yeterli olurdu.
 - ~ Ancak, bilgisayarlar her geçen gün hızlanmakla birlikte sonsuz hıza sahip değildir. Ayrıca sistem kaynakları da ucuzlamakla birlikte sınırsız değildir.
 - ~ Bu sebeple zamanı ve kaynakları verimli kullanmalıyız.

Algoritma

- Algoritmik verimlilik oldukça önemli bir konudur.
 - Farklı algoritmalar aynı problem çözerken verimlilik açısından farklılık gösterebilir.
 - İleriki bölümlerde sıralama probleminin farklı algoritmalar ile çözülebildiğini göreceğiz.
- ~ Bu algoritmalarından biri eklemeli sıralama (insertion sort), kabaca $c_1 n^2$ zamanda n elemanı sıralayabilmektedir. Formüldeki c_1 değeri sabittir ve n sayısına bağlı değildir.
 - ~ Diğer bir algoritma olan birleştirmeli sıralama (merge sort), kabaca $c_2 n \lg n$ süre almaktadır. Formüldeki $\lg n$, $\log_2 n$ manasına gelmektedir ve c_2 değeri sabittir, n sayısına bağlı değildir.
 - ~ Eklemeli sıralama genellikle birleştirmeli sıralamaya göre daha küçük bir sabite sahiptir, yani $c_1 < c_2$.
 - ~ Bu bölümde göreceğimiz üzere sabitlerin algoritmanın toplam çalışma süresine etkisi, sıralanacak eleman sayısına göre daha azdır.

Algoritma

- Bu analizi yapabilmek için eklemeli sıralamanın toplam çalışma süresini $c_1 n^2$, birleştirmeli sıralamanın çalışma süresini ise $c_2 n \lg n$ olarak yazalım.
- Bu iki formülü karşılaştırdığımızda eklemeli sıralama n faktörüne sahipken, birleştirmeli sıralama $\lg n$ faktörüne sahip olduğunu görüyoruz.
 - ~ $n = 1000$ olduğunda $\lg n$ yaklaşık olarak 10 olmakta, $n = 1,000,000$ olduğunda ise $\lg n$ yaklaşık olarak 20 olmaktadır.
- Yukarıda belirtilen sebepler nedeniyle eklemeli sıralama az sayıda elemanın sıralanması durumunda birleştirmeli sıralamaya göre daha az sürede sonuca ulaşırken eleman sayısı arttıkça birleştirmeli sıralama daha hızlı sonuç vermeye başlamaktadır.

Algoritma

- Örnek: Bilgisayar A saniyede 1,000,000,000 komut gerçekleştirebiliyor.
- Bilgisayar B ise saniyede 10,000,000 komut gerçekleştirebiliyor.
- Her iki bilgisayar 1,000,000 sayı içeren bir diziyi sıralamak istiyor.
- Bilgisayar A eklemeli sıralama kullanırken, bilgisayar B birleştirmeli sıralama kullanıyor.
- Ekelemeli sıralama için c_1 değeri 2, birleştirmeli sıralama için c_2 değeri 50 kabul edelim.
- Hangi bilgisayar daha çabuk bitirir?
- Çözüm tahtada gösterilmiştir.

