

BÖLÜM 3. ALGORİTMA TASARIMI

Bu noktaya kadar bir algoritmanın ne olduğu ve taşıdığı özellikler açıklandı. Bununla birlikte algoritmaların zaman bakımından performans kriteri olan $T(n)$ zaman bağıntısının basitçe elde edilişi gösterildi. Bu kısımda ise algoritma tasarım yöntemlerinin kısa bir hikayesi verilecektir. Sekiz değişik algoritma tasarım yöntemi vardır.

- a) Bilinen probleme indirgeme
- b) Özdevinim
- c) Bir veri yapısı icat etme
- d) Böl ve yönet
- e) Kaba Seçim (Greedy) algoritması
- f) Dinamik programlama
- g) İhtimali (olasılıksal) çözümler
- h) Yaklaşım çözümleri

Bu paradigmlar ileri ki konularda daha detaylı olarak inceleneceklerdir.

Kompleks olan bir problemin çözümünü yapmak için çözümü bilinen bir veya birden fazla başka probleme dönüştürüp bu şekilde problemi çözmeye işlemine **bilinen probleme indirgeme yöntemi** denir.

Problemin tabiatından faydalanarak çözümün tekrarlı olması (bilinen bir veya birkaç çözümden faydalanarak bir sonraki çözümü elde etme ve elde edilen çözüm ile önceki çözümlerin birkaçının kullanılması ile bir sonraki çözümün elde edilmesi ve bu şekilde devam eden problem çözümleri) durumunda **özyineleme yöntemi** denir.

O ana kadar var olamayan bir veri yapısının icat edilmesi ile problemin çözülmesine **veri yapısı icat etme yöntemi** denir.

Kompleks problemlerin bir bütün olarak çözümleri çok zor olacağından dolayı, bu problemler alt problemlere bölünürler. Bu bölünme işleminin yapılabilmesi için alt problemlerin bir üst

seviyedeki problem ile aynı özelliği sağlamalıdır. Bu yöntem ile algoritma tasarımı yapmaya **böl ve yönet yöntemi** denir.

Optimizasyon problemlerinin çözümü için yerel optimumların seçilmesi ilkesinden yola çıkar ve veriyi belli bir kritere göre düzenledikten sonra ilk veri her zaman optimum çözüme götürür mantığına sahiptir. Temel amaç en iyi sonucu elde etmek için en iyi ara adım çözümlerini seçmeye yönelik bir yöntem olduğundan bu yöntem **haris algoritması yöntemi** denir.

Böl ve yönet yöntemine benzer olarak alt problemlerin çözümlerini birleştirerek çözüme gitme mantığına sahip olup alt problem tekrarı varsa, bunlardan bir tanesi çözülür ve bu çözüm diğer tekrarlar da kullanılır. Bu yöntem ile yapılan algoritma tasarım yöntemine **dinamik programlama yöntemi** denir.

Bazı durumlarda gelişigüzellik ilkesi ile etkili bir şekilde problem çözümü yapılabilmektedir. Bunlara örnek olarak Las Vegas polinom-zamanlı ve Monte Carlo polinom-zamanlı algoritmalar verilebilir. Gelişigüzellik kullanılarak yapılan problem çözümlerine **ihtimali çözümler yöntemi** denir.

Çözümü deterministik Turing makinası ile yapılamayan problemler vardır. Bu problemlerin bir kısmına bazı kriterler uygulayarak yaklaşım mantığı ile çözüm üretilebilmektedir. Bundan dolayı bu mantık ile yapılan algoritma tasarımına **yaklaşım çözümler yöntemi** adı verilir.

Bazı kaynaklarda karşılaşılabilecek diğer bir yöntem de (yukarıda verilen yöntemlerden ayrı bir yöntem değil) algoritmanın işleyişi sırasında çözümün yapısında meydana gelen değişime göre isimlendirilen bir yöntem vardır. Bu kısımda örnek olarak kullanılan ArayaEkleSırala() algoritmasının izlemiş olduğu mantık artımsal yaklaşımdır. Çünkü algoritmanın çalışması sürdükçe verinin boyutu artmaktadır.

Bu bölümde ArayaEkleSırala() algoritmasının tasarımının yapıldığı yoldan çok farklı olan böl ve yönet mantığından biraz daha söz edilecek olursa, böyle bir mantığı tarihten hatırlanacak olursa, Avrupa

devletleri Osmanlı İmparatorluğu için denemek istedikleri bir yöntemdi ve hala günümüzde bu yöntem bir çok ülke tarafından diğer ülkeler üzerinde kullanılmaktadır. Çünkü yönetilecek kümenin boyutu küçüldükçe, yönetim işi basitleşir.

Bilgisayar mühendisliğinde, bu mantık veriler kümesinin boyutunun küçülmesi işlemleri basitleştireceğinden dolayı kullanılmaktadır.

Eğer bir dizinin elemanları sıralanacaksa, bu dizinin elemanları iki parçaya bölünür ve her parça kendi içinde sıralandıktan sonra iki tane sıralı parça elde edilmiş olur. Bu iki sıralı alt dizinin birleştirilmesi işleminin zaman bağıntısı lineer olur. Bunun anlamı daha kısa bir sürede birleştirme işlemi gerçekleştirilir. Alt dizilerin sıralanıp sıralanmayacağı alt dizinin içindeki eleman sayısına bağlıdır. Eleman sayısı 1 veya 2 oluncaya kadar bölme işlemine devam edilir. Eğer bir parçada bir tane eleman kaldıysa, bir eleman kendi içinde zaten sıralıdır ve eğer iki tane eleman kaldıysa, bu elemanlar sıralanır ve bir sıralı alt dizi elde edilmiş olur.

Bir problemin çözümü yapılırken, böl ve yönet stratejisine eğer problem alt problemlere bölündüğünde alt problemlerde ana problemin göstermiş olduğu bütün özellikleri gösteriyor ve yapısal olarak ana problemten farklı bir özellik göstermiyorsa, bu tip problemlere böl ve yönet stratejisi uygulanabilir. Bu strateji algoritma tasarımında çok güçlü stratejilerden biridir. Daha sonraki bölümlerde böl ve yönet stratejisi ile tasarımı yapılmış algoritmalar üzerinde durulacaktır.