**ALGORİTMAYI NEDEN ANALİZ EDERİZ? ALGORİTMA ANALİZİNİ NASIL YAPARIZ?**

**ALGORİTMAYI NEDEN ANALİZ EDERİZ?**

Algoritma analizi, algoritmanın yürütülmesi için gerekli kaynak miktarının belirlenmesidir. Belirli bir problemi çözen herhangi bir algoritmanın ihtiyaç duyduğu kaynaklar için teorik tahminler sağlar. Başka bir ifadeyle, algoritmanın performansı ve kaynak kullanımı konusunda yapılan teorik çalışmaların tümüne algoritma analizi denir.

Büyük bir miktarda veriyi işlerken bunu olabildiğince hızlı yapmak isteriz. Buradaki hızlı ve yavaş kavramları, girdi boyutu ile doğrudan ilişkilidir. Örneğin 10 tane veriyi 1000 tane veriye göre çok daha hızlı sıralayabiliriz ama bu, algoritmanın daha hızlı olması için daha az girdiye sahip olması gerektiği gibi saçma bir sonuç çıkarmamalı. Yani girdi miktarını örneklendirerek kıyas yaparız ve bir tahminde bulunuruz. Çünkü işlenmesi gereken veri miktarı çoğu zaman kesin olarak bildiğimiz bir şey değildir. İşte bu yüzden, emin olamadığımız sayıda veri girişi içeren bir algoritmayı tasarlamak veya tercih etmek için analize ihtiyaç duyarız.

Programlamada problem çözme kabiliyeti, yapılan işin kalitesi açısından çok önemlidir. Bir problemin çözümü için gelişigüzel mantıklar kurmak ya da aynı şekilde rastgele algoritma tercihi yapmak programın kalitesinden ödün vermek demektir. Evet, belirli bir problemi farklı algoritmalar ile pek tabii çözebiliriz ancak o probleme özel olan en uygun çözümü getirecek algoritmayı belirlemek, belki de problemi çözmekten çok daha önemlidir ve gereklidir.Çünkü günün sonunda algoritma bir şekilde oluşur ve sorun çözülür ama her bakımdan maksimum performansı elde eden tek bir algoritma vardır.İşte o algoritmayı bulmak, diğer algoritmalarla arasındaki farkları ve avantajları belirlemek veya kısaca kıyas yapmak için algoritma analizine ihtiyaç duyarız. Bazen de karşılaştırma yapmaktan ziyade sadece bir algoritmanın performansını ölçmek için analize ihtiyaç duyarız.

**ALGORİTMA ANALİZİNİ NASIL YAPARIZ?**

Algoritma analizinde 2 temel kıstas üzerinden değerlendirme yapılır;

* Süre Karmaşıklığı
* Bellek Alan Karmaşıklığı

Zaman karmaşıklığı bir algoritmanın asimtotik notasyona göre karmaşıklığının ifadesidir. Bellek alan karmaşıklığı ise bellek gereksiniminin karmaşıklığını ifade eder.

Yürütme zamanı, ‘n’ boyutlu bir problemin algoritmasını çalıştırmak için gerekli olan zamana denir. Başka bir ifadeyle karşılaştırma, döngü çevirimi, aritmetik işlemler gibi algoritmanın işlevini yerine getirmesi için temel kabul edilen işlemlerin kaç kere yürütüldüğünü veren bir bağıntıdır.

Yürütme zamanı denkemi üzerinden algoritmanın çalışacağı en kötü, en iyi ve ortalama durumu belirlenir. En kötü durumuna O Notasyonu, En iyi duruma Omega Notasyonu, Ortalam duruma ise Teta Notasyonu denir.