Bilgisayar programcılığını öğrenmenin en zevkli yolu: Algoritma ve Akış Şeması Oyunu

Yıllardır bilgisayarla uğraşıyorsunuz ve geriye bir tek bilgisayar programcılığı kaldı diyorsanız o halde gelin beraber bir oyun oynayalım ve bilgisayar programcılığının abecesi olan algoritmayı öğrenelim.

Sakın korkmayın! Size, bütün çağların en korkunç dersi matematiğin son yılların en keyifli uğraşı olan bilgisayarı yarattığını söylemeyeceğim. Daha farklı bir şekilde ifade etmek gerekirse despot ve keskin kurallara sahip bir baba olan matematik ile güzel, çekici, gizemli ve bir o kadar da hassas kızı bilgisayar arasındaki duygusal bağı anlatacağım. Son dönemlerde filmlerde de gördüğümüz şekliyle bilimin en karmaşık problemleri artık sadece en güçlü bilgisayarlar ile çözülebiliyor. O halde tavuk-yumurta bilmecesine matematik-bilgisayar ilişki de eklenebilir. Matematik ile bilgisayar arasındaki ilişkiyi gerçekten anlayabilmek için bilgisayarların elektronik devrelerindeki yarı-iletken teknolojisini de anlatmak isterdim ama anlatırken ben, okurken de siz sıkılırdınız. En ilgi çeken yönüyle bir problemin bilgisayar programını hazırlamak daha zevkli olacaktır. Bunun için ilk adım "algoritma" denilen tekniği tanımak olmalıdır.

Matematikten bilgisayara giden yol: Algoritma

Bilgisayar kullanımının yaygınlaşmasıyla beraber çağın en iyi mesleklerinden birisi olarak görülen bilgisayar programcılığına ilgi gün geçtikçe artmaktadır. Bununla beraber ticari amaçlı firmaların bilgisayar programcılığını kolaylıkla ve herkese öğretebilme vaatleri genellikle gerçeği yansıtmamaktadır. Müşteriyi ürkütmemek için programcılık ile matematik ilişkisi kimseye açıklanmamaktadır. Bu yüzden bir programcılık kursunda genellikle algoritma ve matematikten çok az bahsedilir veya adı bile anılmaz. Çok kolay olmasa da algoritma hazırlamak zor değildir. Eğer programcılığa algoritma öğrenmekten başlanmazsa temeli atılmamış bir binanın neden gökdelen olamadığını sorgulayıp durabiliriz. Bu yüzden programcılığa, doğru yerden, algoritmadan başlamalıyız.

Bir kaynağa göre *algoritma (algorithm)* sözcüğü 9. yüzyılda cebir alanındaki çalışmalarını kitaba dökerek matematiğe büyük bir katkı sağlayan el Harezmi isimli bilim adamından kaynaklanır. Dünyanın ilk cebir ve algoritma koleksiyonunu oluşturan eserlerinin Latince çevirisi Avrupa'da çok ilgi görür. Fakat söylenmesi zor geldiği için bilim adamının ismi *algorizm* olarak telaffuz edilir. O dönemde sayısal işlemlerde roma rakamlarını kullanan Avrupalı araştırmacılar aritmetik problemleri Arap sayıları kullanarak çözmeye *algorizm* demişlerdir. Daha sonraki zamanlarda da kelime değişerek *algorithm* şekline dönüşmüştür.

Kelime anlamı olarak algoritma, bir amaç için oluşturulan yöntemin sonuç elde edilinceye kadar sürdürülmesidir. Başka bir deyişle algoritma, belirli bir kurala bağlı olarak yapılan her türlü hesaplamaya verilen isimdir. Matematikte ise algoritma bir sorunun yanıtını ya da bir problemin çözümünü belirli sayıdaki aşamalar sayesinde veren sistematik bir yöntemdir. Algoritma, öngörüsel çözüme ve deneme-yanılmaya karşıt olan bir yöntemdir. Algoritmalar özel durumlara çözüm sunmazlar, genel çözümlerin işlem adımlarını içerirler.

Hayatı kolaylaştırmak için geliştirilen bilgisayarların temel amacı doğadaki problemleri çözebilmektir. Problemlerin bilgisayarlara aktarılmasında algoritmalar bir köprü konumundadır. Bir problemin bilgisayar modelini oluşturabilmek için önce problemi anlayarak analiz etmek ve çözüm yollarını ortaya koymak gerekir. Diğer bir deyişle problemin ve çözümün algoritması geliştirilir. Bir bilgisayar, problemin nasıl çözüleceği konusunda insanlara yardımcı olamaz. Sadece insanların çözüm için gösterdiği yollardan giderek istenenleri hatasızca uygular. Bilgisayarın doğru sonuca ulaşabilmesi için kendisine gösterilen çözüm yolunda hiçbir belirsizlikle karşılaşmaması gerekir. Bu nedenle hazırlanacak algoritmada her türlü detay önceden düşünülmüş ve karşılaşılabilecek değişik durumlarda bilgisayarın çözüme nasıl devam edeceği bildirilmiş olmalıdır. Bir algoritma temel olarak sonluluk, kesinlik ve genellik özelliklerine sahip olmalıdır.

Bir probleme ait bilgisayar programının hazırlanabilmesi için aşağıdaki adımlara dikkat edilmelidir.

1. Problemi tanımlama

Her şeyden önce çözülecek problem tam olarak anlaşılmalıdır. Yanlış anlaşılmış bir problemin çözümü yanlış olacak ve istenileni veremeyecektir. Bu adımda yapılacak en ufak bir hata daha sonraki adımların yeni baştan yapılmasını gerektirebilir. Problemin tanımı yapılırken verilen bilgilerin anlamları ve birbirleri ile ilişkileri iyi düşünülmelidir. Daha sonra istenenler belirlenmeli ve bunların verilen bilgiler ile ilişkileri keşfedilmelidir. Son olarak sonuca ulaşmak için yapılacak ara işlemler belirlenmelidir.

2. Algoritma geliştirme

Algoritma bir problemin çözümü için izlenecek yolun tanımıdır. Problem tanımını tam olarak yaptıktan sonra çözüm için yol aramak gerekir. Genellikle bir problemin birden fazla çözüm yolu olabilir. Bunlardan en uygunu seçilmeye çalışılır. Problem ne kadar karışık olursa olsun alt parçalara ayrılabilir. Problem alt parçalara ayrılarak her bir alt parçanın çözümü ayrı yapılır. Dikkat edilmesi gereken şey parçalar arasındaki ilişkinin korunmasıdır. Bu adım programcının deneyimine kalmıştır. Programcılık, için yetenek değil deneyim kelimesinin kullanımı daha uygundur. Çünkü sürekli programlamayla uğraşan birisi için daha önce öğrendiği algoritmaları birleştirip başka bir problemin çözümüne uygulamak çok basit bir iştir.

3. Değişkenler ile girdi ve çıktı biçimi belirleme

Belirli bir tanım aralığında farklı değerler alabilen sembollere değişken denmektedir. Bilgisayar, işlem yaparken RAM belleği kullanır. Program çalıştırılırken problemin çözümü için gerekli olan bilgilerin RAM bellekte tutulması değişkenler sayesinde mümkündür. Değişkenlerin RAM bellekte bulunuşları aynen bir sinema salonunda koltuklara film bitene kadar oturmuş seyirciler gibi düşünülebilir. Aynen sinemada bilet alındığında koltuk numarası ile ilgili yere ulaşılabildiği gibi bir değişken tanımlandığında da RAM bellekte bir yer ayrılır ve bu yere değişkenin ismiyle ulaşılır. Program içinde kullanılacak olan değişkenler problemin yapısına göre algoritma tasarlama aşamasında belirlenmelidir. Değişken isimlerini tanımlarken (veya ilk kullanımında) Türkçeye özgü karakteri kullanmamaya dikkat edilmelidir. Sonuçların dış ortama (kullanıcıya) aktarımı en uygun biçimde yapılmalıdır.

4. Akış şemasını çizme

Akış şeması, kısaca algoritmanın şekillerle gösterilmesidir. Algoritma geliştirildikten sonra hem anlaşılması hem de seçilen programlama diline kolay aktarılabilmesi için akış şeması haline getirilir. Böylece problemin çözüm basamakları, birbirleri ile ilişkileri ve bilgi akışı daha kolay görülebilir ve yanlışlıklar düzeltilebilir.

5. Kodlama

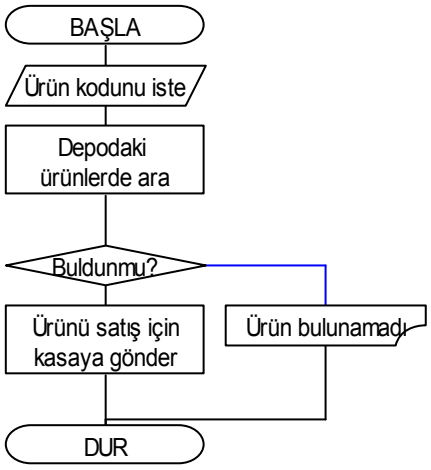
Kodlama için kullanılacak programlama dilinin hangisi olduğu çoğunlukla önemli değildir. Problemde yeni teknolojiye bağımlı bir gereksinim yoksa eski bir programlama dili bile kullanılabilir. Yine de problemin yapısına uygun bir programlama dili seçilmesi gerekebilir. Akış şemasındaki her şematik gösterimin seçilen programlama dilinin bir veya birkaç komutuna karşılık geldiği söylenebilir. Bu yüzden eğer problemin akış şeması doğru çizildiyse problemin programlanması çok kolay olacaktır.

6. Programı sınama

Program kodlandıktan sonra sonuçları daha önceden bilinen veriler girilerek karşılaştırılır. Bu kontrol aşaması çok uç örnekler için tekrar edilmelidir. Çünkü bir algoritma, problemin geneline çözüm üretmelidir.

Algoritmayı şekillerle ifade etme oyunu: Akış Şeması

Akış şeması, bir işin başlangıcından sonlandırılmasına kadar atılması gereken adımların şematik bir biçimde gösterilmesidir. İş akışının grafiksel gösterimi ilk kez Frank Gilbreth tarafından 1921'de kullanılmıştır. Daha sonra endüstri mühendisliğinde Gilbreth'in araçları oldukça benimsenmiştir. Temel olarak başlangıç, bitiş, yapılması zorunlu olan işlemler, bazı durumlarda karşılaşılan ikilemler ve işin akış yönünü gösteren oklardan ibarettir. Akış şemaları günümüzde çeşitli alanlardaki işlem ve programların yönetilmesi, raporlanması, tasarlanması ve analiz edilmesinde yaygın bir şekilde kullanılmaktadır. Aşağıda, ürün koduna göre telefonla sipariş alan bir mağazanın işleyişi şematik olarak gösterilmiştir.



Şekil 1. Telefonla sipariş alan bir mağazanın işleyişi

Diğer alanlarda kullanılan akış şemalarından farklı olarak bilgisayar ortamında matematiksel bir problemin çözümünü tanımlayan akış şemalarında basit cümleler yerine matematiksel ifadeler, denklemler ve işlemler kullanılmalıdır.

Bir bilgisayar programının oluşturulmasında akış şemalarının hazırlanması algoritma oluşturma aşamasından sonra gelmektedir. Tabii ki algoritma aşaması kağıda dökülmeden doğrudan akış şemalarının hazırlanmasına başlanabilir. Programlama tekniğinde önemli ölçüde yol almış kişiler algoritma ve akış şeması hazırlamayı atlayarak programın yazımına geçebilirler. Bu, algoritma aşamasını atladığımız anlamına gelmemelidir. Akış şemalarının algoritmadan farkı adımların simgeler şeklinde olması ve adımlar arasındaki ilişkilerin oklar ile gösterilmesidir. Akış şeması hazırlarken kullanılabilecek şekiller aşağıda verilmiştir.

Tablo 1. Akış şeması için kullanılan semboller

Akış Şeması Nesnesi	**Sembolik Gösterimi**
Başla	BAŞLA
Dur	DUR
Veri girişi	X,Y,Z
İşlem	X=Y+Z
Döngü	i=1,10,1
Karşılaştırma	X>Y+Z
Yazdırma	X,Y,Z

Daha iyi anlayabilmek için aşağıda basit matematiksel işlemleri kullanarak oluşturulan örnek problemleri inceleyelim.

Örnek 1: Kullanıcı tarafından verilen iki sayının toplamını ekrana yazdıralım;

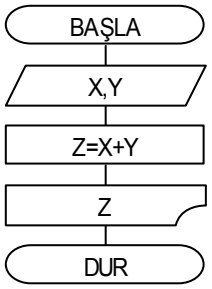
Çözüm: Öncelikle aklımızda belirdiği şekliyle problemin adımlarını yazalım. İki değişken kullanarak bu iki değişkenin içeriklerini dışarıdan istemeliyiz. Örneğin bu iki değişken X ve Y olabilir. X ve Y değişkenlerini toplayıp Z gibi üçüncü bir değişkene aktarmalıyız. Z değişkeninin değerini ekrana yazdırmalıyız.

Şimdi açık şekilde yazdığımız bu adımları biraz daha basit ve terimsel olarak ifade edelim.

1. Kullanıcıdan X ve Y değişkenlerini iste.
2. Toplamlarını hesapla. Z = X + Y

3. Z değişkenini ekrana yazdır.

Şimdi de akış şemasını çizelim.



Şekil 2. Örnek 1 için akış şeması

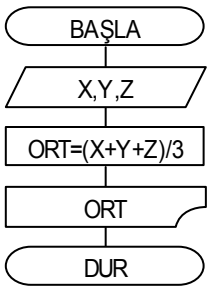
Örnek 2: Dışarıdan girilen üç sayının ortalamasını ekrana yazdıralım;

Çözüm: Problemin adımlarını anlaşılabilir bir dille yazalım. Üç değişken kullanmalıyız ve bu değişkenlerin içeriklerini dışarıdan istemeliyiz. Örneğin bu değişkenler X, Y ve Z olabilir. Bu üç değişkenin ortalamasını bulmalıyız. X, Y ve Z değişkenlerinin toplamını bulup 3'e bölersek ortalamasını bulmuş oluruz. Bulduğumuz ortalama için de bir değişken kullanmalıyız (ORT olabilir). ORT değişkenini ekrana yazdırmalıyız.

Şimdi de adımları biraz matematiksel hale getirelim.

1. X, Y ve Z değişkenlerini iste.
2. Ortalama bul. ORT = (X + Y + Z) / 3
3. ORT değişkenini yazdır.

Şimdi de şematik olarak göstermek için akış şemasını çizelim.



Şekil 3. Örnek 2 için akış şeması

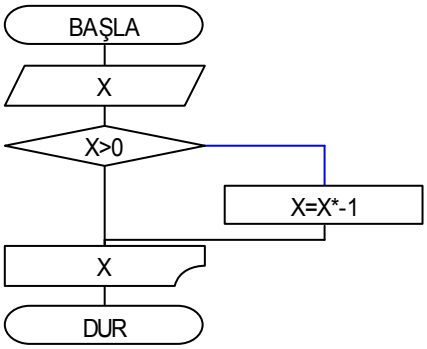
Örnek 3: Dışarıdan girilen sayının mutlak değerini ekrana yazdıralım;

Çözüm: Öncelikle aklımızda belirdiği şekliyle problemin adımlarını yazalım. Tek değişken kullanmalıyız. Bu değişkeni dışarıdan istemeliyiz. Örneğin bu değişken X olabilir. Eğer X değişkeninin değeri sıfırdan büyükse bir sonraki adıma geçebiliriz, ama değilse değişkeni -1 ile çarpıp pozitif sayı haline getirmeli ve bu çarpım değerini yine X değişkenine aktarmalıyız. X değişkeninin değerini ekrana yazdırmalıyız.

Şimdi açık şekilde yazdığımız bu adımları daha matematiksel olarak ifade edelim.

1. X değişkenini iste.
2. Eğer X > 0 ise 4. adıma git. (Karşılaştırma nesnesinin iki çıkışı vardır. Karşılaştırma işleminin sonucu doğru ise temsili olarak en alttaki köşe noktası, yanlış ise sağ veya sol köşe noktası kullanılmıştır.)
3. Değişkeni pozitif yap. X = X * -1
4. X değişkenini yazdır.

Algoritmanın akış şemasını çizelim.



Şekil 4. Örnek 3 için akış şeması

Şimdi algoritma oyununu oynama sırası sizde. Birinci problemde algoritmanın açık ve matematiksel adımlarını yazıp sizden akış şemasını istiyorum. Problem matematik derslerinden bildiğimiz asal sayı problemi.

Problem 1: Dışarıdan girilen sayının asal sayı olup olmadığını tespit edip ekrana "ASAL" veya "DEĞİL" yazdırmanız isteniyor.

Çözüm: Öncelikle problemin açık adımlarını yazalım. Asal sayılar derslerde "1 ve kendisi dışında hiçbir sayıya tam bölünemeyen sayılar" olarak tanımlanırlar. O halde dışarıdan girilen X sayısının 1 ve X sayısı dışında herhangi bir sayıya tam bölünüp bölünmediğini bulmalıyız. X sayısını tam bölebilecek sayılar mantıken 1'den büyük ve X'den küçük sayılardan bazıları olabilir. O halde 2'den başlayıp X-1 sayısına kadar tüm sayıları tek tek X bölünen sayısına bölen olarak denemeliyiz. Bölme işleminde kalan 0 (sıfır) olduğu zaman tam bölünme vardır ve dolayısıyla X sayısı asal değildir diyeceğiz. Matematikte bölme işlemindeki kalanı veren + (toplama) ve * (çarpma) gibi bir operatör vardır. Genellikle "mod" adıyla bilinen bu işlem operatörü için % karakterini kullanacağım. Dolayısıyla 10%4 denildiğinde 10 sayısının 4'e bölümünden kalanı verecek bir işlemi tanımlamış olacağız. Bir döngü içinde X sayısını 2'den X-1'e kadarki tüm sayılara "mod"unu hesaplamalıyız. Eğer kalan değeri 0 (sıfır) ise X sayısının asal olmadığını anlayacağız. Eğer döngü bitene kadar hiçbir sayı tam bölen olamazsa X sayısı asaldır diyeceğiz. O halde döngü bittiğinde bunu anlamamızı sağlayacak bir değişken kullanmalıyız. Bu değişken A olabilir. A değişkenini döngü başlamadan bir değer ile sabitleriz ve döngü bittiğinde de aynı değerde duruyorsa asaldır deriz. Döngü içinde de "mod" işlem sonucunu k diye bir değişkene aktarıp bu değişkenin değeri 0 (sıfır) ise A değişkenini başka bir değere eşitleriz.

Şimdi açık şekilde yazdığımız bu adımları daha matematiksel olarak ifade edelim.

1. X değişkenini iste
2. A değişkenini başlangıç değerine sabitle. Örneğin 1 olabilir. A = 1
3. Bir döngü kur. Döngü değişkeni 2'den başlayıp birer birer X-1'e kadar artsın. i=2,X-1,1 (Bu ifadede i döngü değişkeni, 2 ilk değer, X-1 son değer ve en sondaki 1 ise döngü değişkeninin ilk değerden son değere kadar artış değerini göstermektedir. Döngü değişkeni son değeri de kullandıktan sonra 8. adıma gidecektir.)
4. Döngü içinde X sayısının i sayısına "mod"unu (bölümünden kalanı) hesapla. k=X%i
5. Eğer k değeri sıfır değil ise 7. adıma git.
6. A değerini değiştir. A = 0
7. Döngüye devam etmek için 3. adıma git.
8. Eğer A değeri sıfır ise 10. adıma git.
9. Ekrana "ASAL" yazdır.
10. Ekrana "DEĞİL" yazdır.

Bu algoritmanın akış şeması için şimdi sıra sizde. Kolay gelsin.

Kaynaklar

1. Nabiyev, V. V., *Algoritmalar Teoriden Uygulamalara*, Seçkin Yayıncılık, 2007.
2. Çobanoğlu , B., *Algoritma Geliştirme ve Veri Yapıları*, Pusula Yayıncılık ve İletişim, 2009.
3. Yaşar, E., *Algoritma ve Programlamaya Giriş*, Murathan Yayınevi, 2005.
4. http://www.gop.edu.tr/umutorhan/dersnotu/Algoritma_Ders_Notu.pdf