

ALGORİTMALAR VE AKIŞ DİYAGRAMLARI

Mustafa NAR*

Sayılar günlük hayatımızın vazgeçilmez unsurlarıdır. Mazisi ilk insanla aynı tarihi paylaşır. Geçmişte insanlar sahip oldukları eşya ve malların miktarını bilmek için sayılarla işlemler yaptı. Böylece Matematik, insanların uğraştığı en eski bilim dallarından biri oldu. Her şeyi kafadan hesaplamanın zor olduğunu gören insanlar, hesaplamada yardımcı olacak araç ve gereçleri geliştirmede gecikmedi. Meselâ, Roma devrinde kafadan hesaplamaya yardımcı olmak için parmaklar kullanıldı. Sonraları, yaklaşık olarak günümüzden 2300 yıl önce Çinliler tarafından "ABAKÜS" adlı araç yaygın olarak kullanılmaya başlandı. Bu araç bazı geri kalmış ülkelerde hâlâ kullanılmaktadır.

Abaküs ile hesaplamada eldeleri kafada tutma zorunluluğu vardır. Eldeleri kullanarak toplama yapabilen ilk alet, 1642 yılında Fransız Blais Pascal tarafından icad edildi. 1671 yılında sadece çarpma işlemini yapabilen bir makine Leibniz tarafından geliştirildi. Bu çalışmalardan yararlanarak Avrupa ve Amerika'da dört işlem yapabilen masa hesap makineleri imal edildi.

Dört işlemi yapan bu makinelere bir hafıza eklenmesi üzerine yapılan çalışmalar semeresini ancak 1890 yılında verdi. O yılda ABD'de nüfus sayımında sonuçların çabuk alınması için Hollerith tarafından kendi adıyla anılan delikli kart makineleri yerleştirildi. Bu çalışmalar günümüzün modern bilgisayarlarına doğru atılan önemli bir adımı teşkil eder.

İlk bilgisayarlar askerî amaçlı ve sadece onun için yapısını bilen bilim adamları tarafından kullanılabiliyordu. Bilgisayarların kullanılması yaygınlaşınca, bilgisayarın, onun iç yapısını ve aksamlarını (Hardware) bilmeyenlerin de kullanabilmesi için, kullanıcıların da anlayacağı programların dilleri geliştirildi. Böylece yazılım (Software) ortaya çıktı. Yazılım, algoritmalar, program akış şemaları, işletim talimatı vs. gibi bilgisayara yapacağı işin tarif edilmesinde kullanılan unsurlardır.

Günümüzde birden fazla bilgisayar dili bulunmakta ve hepsi de değişik amaçlar için kullanılmaktadır. Bunlar, makine dilleri, Asembler dilleri ve daha çok insanların anlayacağı dile yakın olan sembolik



programlama dilleri (BASIC-COBOL-FORTRAN-PL/PASCAL vs.) olarak sayılabilir.

Bir programcı, bilgisayara çözdürmek istediği problemin mantığını iyi ve doğru kurmak zorundadır. Aksi halde, bu yanlış mantık üzerine yazılan program dili, bilgisayara, sonucu yanlış bir işlem yapacaktır. Meselâ; biz, bir programı yazarken $X = 5$ ve $Y = 5$ bunu da $X + Y = 12$ şeklinde tanımlayıp bilgisayara verdiğimizde yanlış sonuç alırız. Bunun sorumlusu da bilgisayar değil, programı yazan biz ya da bilgileri yükleyen olacaktır. O halde problemi çözmek için yazılacak herhangi bir sembolik dilden önce, çözülecek problemin çok iyi anlaşılması, çözüm sırasında kullanılan tüm adımların bilinmesi gerekmektedir. Çünkü çözdüreceği problemin unsurlarını, adımlarını bilmeyen biri, bunun programını doğru olarak yazıp bilgisayara çözdüremeyecektir. Bilgisayarlar kendilerine verilen programı yorumlarlar, kitap okur gibi satır satır, adım adım yorumlarlar. Programın mantığına göre, meselâ 35. satırda iken, bir emir ile 100. satırdaki emirleri yerine getirip tekrar 36. satırdan yorumlamaya devam edebilir. Ya da 40. satırda iken, bir durum değerlendirmesi yapıp, 70. veya 80. satıra atlayıp, satır satır emirleri yorumlamaya devam eder.

Bilgisayarlarda hangi programlama dili kullanılırsa kullanılsın, algoritma ve akış diyagramları aynıdır. Çözülmesi istenen problemin çözüm mantığı tesbit edildikten sonra bunun bilgisayarda, sözlü olarak, adım adım nasıl çözüleceğini anlatan yazılıma ALGORİTMA denir.

Verilen iki sayıyı çarpıp sonucunu yazan bir programın algoritmasını yazmak isteyelim. Bunun için, bu iki sayıyı temsil eden değişkenleri tanımlamak gerekecektir. Böylece bilgisayara, ilk sayının A, ikinci sayının B, ikisinin çarpılması ile elde edilen değer ise C değişkenine aktarılacağını belirtmemiz gerekecektir. Bilgisayar, bu çarpma işlemini yapmadan önce, A ve B olarak temsil edilen sayıları tanımlıdır. Demek ki, bir bilgisayar yapacağı işlemin unsurlarını, değişkenlerini önceden tanımalı, bilmelidir. Yukarıdaki işlemi yaparken bilgisayarın takip etmesi gereken işlem, adım adım

* Elektronik Mühendisi

1. Başla
2. A'yi tanı
3. B'yi tanı
4. $C = A \times B$
5. C'yi yaz
6. Dur.

olacaktır.

Eğer yukarıdaki işlemi,

1. Başla
2. $C = A \times B$
3. C'yi yaz
4. A'yi tanı
5. B'yi tanı
6. Dur.

şeklinde yaptırmak isteseydik, bilgisayar bu işlemi yapamayacaktı. Çünkü, bilgisayarlar çarpma yapmadan önce, birinci sayıyı, sonra ikinci sayıyı tanımalı ve bu iki sayıyı çarpıp 5. adımda da sonucu yazmalıdır.

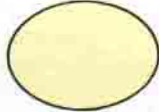
Yine başka bir misal de, bir insanın kabaca yaşını bulan bir programın algoritması,

1. Başla
2. Bu yılı tanı
3. Doğum yılını tanı
4. Yaş = Bu yıl — Doğum yılı
5. Yaş'ı yaz
6. Dur.

şeklinde olacaktır. Yukarıda anlatılan bu ALGORİTMA'nın bir şema halinde gösterilmesine ise AKIŞ DİYAGRAMI denir. Yazılımın akış diyagramı kısmını gerçekleştirmek için yapılacak iş, Algoritmada her adıma denk düşen sembolü çizmek olacaktır. Akış diyagramının kullanılmasının sebebi, sözlü olarak anlatamayacağımız program adımlarının sembolik olarak gösterilebilmesindedir. Böylece, algoritmada sözlü olarak anlatılan programın adımları, akış diyagramında şekillere dönüşecektir. Bu şekiller,

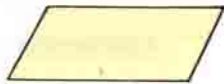
1. BAŞLA veya DUR

Bu sembol akış diyagramının başladığını ve bittiğini gösterir.



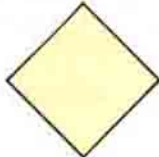
2. BİLGİ GİRİŞİ

Bilgisayara, dışarıdan (klavyeden, kart okuyucudan veya diskten) yüklenecek bilgilerin okutulmasında kullanılır.



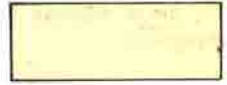
3. KARAR SEMBOLÜ

Bir ya da daha fazla değişkenin birbiriyle mukayese edilip, bunlara göre akış diyagramının yönünün değiştirilmesinde kullanılır.



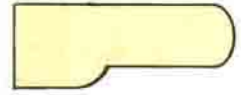
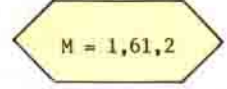
4. İŞLEM SEMBOLÜ

Yapılacak matematik işlemler bu sembolün içerisinde gösterilir.



5. DÖNGÜ

Yapılacak işlem birden fazla tekrar ediyorsa, bunlar bir defa yazılır ve kaç defa tekrar ettiği bu sembol içerisinde değişken adı, sırasıyla, değişkenin alacağı ilk değer, alacağı en büyük değer ve bu değere ulaşması için kaçar kaçar arttırılacağı yazılır. Meselâ, $M = 1, 61, 2$ gibi... Burada M değişkenin alacağı ilk değer 1, alacağı en büyük değer 61 ve bu değere 2'şer 2'şer arttırılarak ulaşacağı gösterilmektedir.

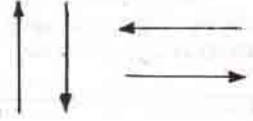


6. ÇIKIŞ SEMBOLÜ

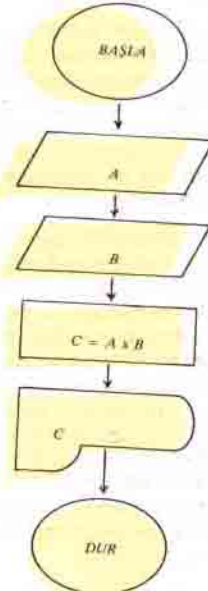
Hesaplanan sonuçların ekrana ya da yazıcıya yazılmasını göstermek için kullanılır.

7. İŞLEM AKIŞ YÖNÜ

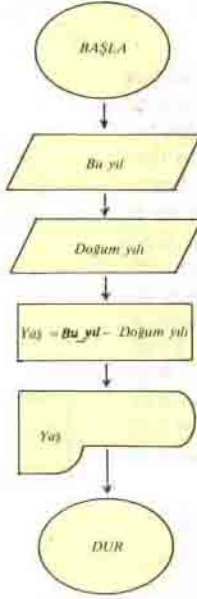
Akış diyagramında işlemin takip edeceği yönleri belirtmek için kullanılır.



Akış diyagramına birkaç örnek vermek için önceki çarpma işlemi yapan algoritmayı kullanalım.



İkinci misalde yaş bulan algoritmanın akış diyagramı :



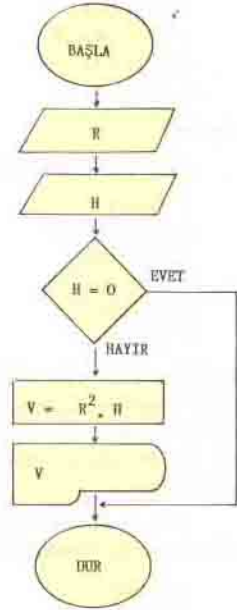
DEĞİŞKENLER:

R : Yarıçap
H : Yükseklik
V : Hacim

ALGORİTMASI:

1. BAŞLA
2. π 'yi tanı
3. R'yi oku
4. H'yi oku
5. Eğer H = 0 ise dur
6. $V = \pi R^2 H$
7. V'yi yaz
8. DUR

AKIŞ DİYAGRAMI :



Yukarıda da belirttiğim gibi, hangi dilde yazılırsa yazılsın, algoritmalar ve akış diyagramları hemen hemen aynıdır. Görüldüğü gibi bilgisayar programcılığının ilk adımı ALGORİTMALAR ve AKIŞ DİYAGRAMLARI'dır. □

SÜPER EKRAN

Bilgisayarlar gün geçtikçe ucuzlayan cihazlar olmalarına rağmen, bunların pahalı katot ışınli tüp (KIT)'e başka bir deyişle ekrana olan bağımlılıkları devam etmektedir. KIT'ler zaman zaman problemlere de sebep olmaktadır. TV'de de kullanılan KIT, okuma-yazma vb. işlemler için elverişli olmayan ortam ışığında, parlaklık ve yansımalarından dolayı iyi bir görüntü sağlayabilmektedir.

KIT'lerin kenarlarının tam köşeli(90°) olmaması sebebiyle, ekranda tam gerçek bir resim görüntülenememekte; bu da tasarımcılar ve kullanıcılar için sıkıntıya sebep olmaktadır. Bu problemi ortadan kaldırmak için Zenith firması, Düz Gerilmeli Maske (DGM) adını verdiği yeni bir resim tüpü geliştirdi. Bu tüp, bilgisayar kullanıcılarına yazıcıdan alınan sayfa görüntüsünün yansırı, aynısını ekranda görme imkânı sağlıyor. Yukarıda şekli görülen Zenith firmasına ait DGM, normal KIT'lere oranla % 50 daha parlak ve yaklaşık % 70 daha fazla görüntü netliğine sahip. Böylece ekran görüntüsünün normal ortam ışığında rahatlıkla görülmesi sağlanmış oluyor. Ayrıca, ekran



kare-köşe olarak tasarlandığından, geleneksel KIT'lerin aksine, doğru çizgiler doğru, daireler ise oval yerine daire olarak görüntülenebilmektedir.

Omni'den çev: Abdullah ÇAVUŞOĞLU