

## Məsələlərin kompyuterdə həlli

Kompyuterdə məsələlərin həlli aşağıdakı mərhələlər ardıcılığı ilə aparılır:

- məsələnin qoyuluşu;
- həll alqoritminin yaradılması;
- verilənlərin strukturlarının təyini;
- proqramlaşdırma dilinin seçilməsi və ilkin proqramın tərtibi;
- proqramın kompyuter dilinə çevrilməsi və sazlanması;
- işçi proqramın icrası, nəticələrin alınması və təhlili.

Müəyyən tip məsələlərin həllində bu mərhələlərdən bəziləri tələb olunmaya bilər. Məsələn, sistem proqram təminatının yaradılmasında məsələnin riyazi təsviri tələb olunmur.

**Məsələnin qoyuluşu.** Məsələnin müvəffəqiyyətli həlli onun düzgün qoyuluşundan asılıdır. Məsələnin qoyuluşu sadə halda aşağıdakıları nəzərdə tutur: ilkin verilənlərin siyahısı, tipi, dəqiqliyi və ölçüləri; dəyişənlərin dəyişmə hədləri, başlanğıc və sərhəd şərtləri; nəticələrin siyahısı, tipi, dəqiqliyi və ölçüləri; məsələnin həllini təmin edən hesabat düsturları və tənlikləri.

**Həll alqoritminin yaradılması.** Bu mərhələdə seçilən həll metoduna uyğun məsələnin həll alqoritmı tərtib edilir. Məsələnin həlli ayrı-ayrı müstəqil bloklara bölünür və həmin blokların yerinə yetirilmə ardıcılığı təyin edilir. Nəticədə alqoritmın blok-sxemi qurulur.

**Verilənlərin strukturlarının təyini.** Bu mərhələdə alqoritmə iştirak edən verilənlərin tipinə, formasına, mümkün qiymətlərinə və aparılan əməliyyatlarına görə onların strukturları seçilir. Yəni verilənlərin tam, həqiqi, simvol və s. tip olması, massiv, yazı, stek, növbə, siyahı, fayl və s. strukturlarla təşkili müəyyənləşdirilir.

**Proqramlaşdırma dilinin seçilməsi və ilkin proqramın tərtibi.** Hazırda proqramlaşdırma üçün müxtəlif dillər mövcuddur. Həll olunan məsələnin xarakterinə, tətbiq olunan kompyuter üçün mövcud olan translyatorlara, proqramçının hazırlıq səviyyəsinə görə proqramlaşdırma dili seçilir. Sonra isə

məsələnin həll alqoritmı əsasında seçilən dildə proqram tərtib edilir. Ona **ilkin proqram** deyilir.

**İlkin proqramın kompyuter dilinə çevrilməsi və sazlanması.** Kompyuter yalnız “öz” dilində ifadə olunan proqramı icra edir. Kompyuterin “öz” dili isə əmrlərdən təşkil olunur. Kompyuter dilində proqram isə alqoritmə uyğun olaraq yazılmış əmrlər ardıcılığıdır. Bu mərhələdə proqramlaşdırma dilində yazılmış ilkin proqram kompyuter dilinə çevrilir. Bu iş translyator adlanan proqram vasitəsilə yerinə yetirilir. Bu zaman ilkin proqramda buraxılmış morfoloji və sintaksis səhvlər aşkar edilib proqramçıya çatdırılır. Səhvlər aradan qaldırıldıqdan sonra tərcümə prosesi davam etdirilir və kompyuter dilində proqram alınır. Bu proqrama **mütləq** və ya **işçi proqram** deyilir. Sonra işçi proqramın düzgün işləməsini yoxlamaq məqsədi ilə yoxlama misalında o, sınaqdan keçirilir. Proqramdakı məntiqi səhvlərin aşkarlanıb aradan qaldırılması prosesinə **proqramın sazlanması** deyilir.

**İşçi proqramın icrası, nəticələrin alınması və təhlili.** Proqram sazlandıqdan sonra ondan tətbiqi məsələnin həlli üçün istifadə etmək olar. Bu zaman proqram müxtəlif ilkin verilənlər dəsti üçün bir neçə dəfə icra olunur. Alınmış nəticələr mütəxəssis və ya məsələni qoyan istifadəçi tərəfindən təhlil olunur. Əgər təhlil prosesində nəticələr istifadəçini təmin etmirsə, o, yeni tələblərini qoya bilər və ya əvvəlki tələblərdə dəyişiklik edə bilər. Bu halda yeni tələblərin xarakterindən asılı olaraq məsələnin qoyuluşunda, alqoritmə və ya proqramda müəyyən dəyişikliklər edilir.

## **Alqoritm və onun xassələri**

**Alqoritm** - verilmiş hər hansı tip məsələnin həlli üçün yerinə yetiriləcək əməliyyatlar ardıcılığıdır. Ümumi şəkildə desək, alqoritm məsələnin həll yoludur, başqa sözlə, məsələnin həllini təmin edən formal qaydalar sistemidir.

Bizim hər bir görəcəyimiz iş müəyyən bir alqoritmə əsaslanır. Məsələn, yeməyin hazırlanma resepti, mebelin quraşdırılması üzrə təlimat, kompyuter proqramlarının istifadəsi üçün dərslik və s. Bu siyahını istənilən qədər artırmaq olar.

## Alqoritmin yaranma tarixi

Alqoritm anlayışı riyaziyyatda eyni tip məsələlərin həllində ümumi metodların axtarılması ilə əlaqədar olaraq meydana çıxmışdır. Çoxrəqəmli onluq ədədlər üzərində hesab əməllərinin aparılması qaydaları (alqoritmləri) ilk dəfə IX əsrin görkəmli özbək riyaziyyatçısı **Əl-Xarəzm** tərəfindən verilmişdir. Alqoritm termini də məhz bu riyaziyyatçının adı ilə bağlıdır (**Algorithmi**). Onun hesab elminə dair yazdığı “Hind rəqəmləri ilə hesablama kitabı” adlı əsərinin latınca tərcüməsi gəlib bizə çatmışdır. Bu kitab “Alqoritm dedi” kəlmələri ilə başlayır. Burada işlədilən “Alqoritm” sözü uzun müddət riyaziyyatçılar üçün riyazi sirr olaraq qalmışdır. Nəhayət, XIX əsrin 40-cı illərində dəqiq müəyyən edildi ki, bu söz “ƏL-XARƏZMİ” sözünün latıncada düzgün olmayan tələffüzü nəticəsində alınmışdır.



Məsələnin kompyuterdə həlli baxımından **alqoritm** axtarılan cavabların alınması üçün məsələnin verilənləri üzərində icra olunan hesab və məntiqi əməllər (mərhələlər) ardıcılığıdır. Bu mərhələlərdə uyğun olaraq hesab və müqayisə əməlləri yerinə yetirilir. Müqayisənin nəticəsindən asılı olaraq digər mərhələnin icrasına keçilir.

Alqoritmdeki hesab əməlləri arasındakı məntiqi əməllər kompyuterin qəbul edə biləcəyi şəkildə verilməlidir. Məntiqi şərtlər içərisində aşağıdakılar xüsusi yer tutur, çünki onların yaranması hesablama prosesinin normal gedişinə imkan vermir:

1. Hesablamada mütləq qiymətə kompyuterdə təsvir oluna biləcək maksimal ədəddən böyük ədədin alınması;
2. Sıfırın və ya mənfi ədədlərin loqarifmlərinin hesablanması;
3. Mənasız hesablamaların aparılmasına cəhd göstərilməsi (məs,  $|x| > 1$  olduqda,  $\arcsin x$  və ya  $\arccos x$ -in hesablanması) və s.

## Alqoritmin əsas xassələri

Alqoritm həll olan məsələnin xarakteri ilə bağlı olduğu üçün onun yaradılmasında ümumi qaydalar yoxdur. Lakin hər bir həll alqoritmi tərtib edilərkən onun müəyyən tələblərə cavab verməsi nəzərə alınmalıdır. Bu tələblərə **alqoritmin xassələri** deyilir.

Alqoritmin əsas xassələri aşağıdakılardır:

- Determiniklik (müəyyənlik)
- Kütləvilik
- Nəticəvilik (sonluluq)
- Diskretlik

### 1. Determiniklik (müəyyənlik)

Alqoritmdeki hər bir mərhələnin məzmunu və mərhələlərin yerinə yetirilmə ardıcılığı müəyyən olmalıdır. Bu, alqoritmin müəyyənlik xassəsini təşkil edir.

### 2. Kütləvilik

Bu xassə iki tələbi nəzərdə tutur:

- a) müəyyən məsələnin həlli üçün qurulmuş alqoritm həmin tiptən olan bütün məsələlərin həlli üçün yararlı olmalıdır;
- b) alqoritm elə təsvir olunmalıdır ki, ondan hamı istifadə edə bilsin.

### 3. Nəticəvilik

Alqoritmdeki mərhələlərin və onları təşkil edən əməliyyatların sayı sonlu olmalıdır ki, onların yerinə yetirilməsi axtarılan nəticəyə gətirib çıxara bilsin.

### 4. Diskretlik

Alqoritmdeki mərhələlərin hər biri sonlu zaman müddətində yerinə yetirilməlidir. Belə diskret zaman müddəti *takt* adlanır.

*Hər bir mərhələ yalnız əvvəlki mərhələnin yerinə yetirilməsindən sonra başlayır.*

## Alqoritmin təsvir üsulları

Alqoritmin təsviri üçün istifadə olunan əsas üsullar aşağıdakılardır:

- Sözlə təsvir (Təbii dillə);
- Alqoritmik dillə təsvir (proqram);

- Qrafik təsvir (Blok-sxem).

## Sözlə təsvir

Sözlə təsvir olunan alqoritmın icrası insan tərəfindən aparılır.

**Misal.** İki ədədin – 5 və 4 ədədlərinin hasilinin tapılması alqoritmına baxaq.

- 5 və 4 ədədləri daxil edilir
- vurma cədvəlindən istifadə edərək  $5 \times 4 = 20$  hasilı tapılır
- 20 cavabı qeyd olunur

Bu üsul alqoritmın kütləvilik xassəsini ödəmədiyindən, az hallarda istifadə olunur. Belə ki, alqoritmın bu təsvir üsulu müəyyən bir məsələnin hamı tərəfindən başa düşülməsini çətinləşdirir, bəzi hallarda mümkünsüz edir.

## Alqoritmik dil

Alqoritmın ən yığcam təsvir vasitəsi alqoritmik dildir. Bu üsul alqoritmın icrasının kompyuter tərəfindən yerinə yetirildiyi halda daha əlverişlidir. Çünki alqoritmik dildə təsvir olunan alqoritm həm də məsələnin ilkin proqramıdır. Lakin bu üsul mürəkkəb alqoritmaların oxunub başa düşülməsini xeyli çətinləşdirir.

**Misal.** Sözlə təsvir etdiyimiz alqoritmı alqoritmik dildə təsvir edək. İki ədədin – 5 və 4 ədədlərinin hasilinin tapılması alqoritmına baxaq.

*alqoritm        / a və b ədədinin hasilinin tapılması*

*verilir   / a, b*

*tapmalı /  $c = a * b$*

*başlanğıc    /  $a = 5, b = 4$*

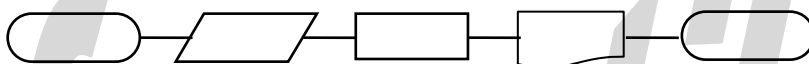
*hesablama   /  $c = 5 * 4$*

*nəticə   /  $c = 20$*

## Qrafik (blok-sxem) üsul

Alqoritmın təsvirində ən geniş tətbiq edilən üsul blok-sxem üsuludur. Bu üsulda alqoritm, hər biri müəyyən funksiyanı yerinə yetirən bloklar ardıcılığı şəklində təsvir olunur. Adətən bir blok alqoritmın bir mərhələsinə uyğun olur. Lakin bir blokda bir neçə eyni tipli mərhələ və ya əksinə, bir mərhələ bir neçə blokda təsvir oluna bilər. Bloklar həndəsi fiqur şəklində ifadə olunur və bir-biri ilə

şaquli, yaxud üfüqi xətlərlə birləşdirilir. Əgər xətlərin uclarında istiqaməti göstərən ox işarəsi yoxdursa, onda keçidin şaquli istiqamətdə yuxarıdan aşağıya, üfüqi istiqamətdə isə soldan sağa verildiyi qəbul olunmuşdur. Lazım gəldikdə bloklar nömrələnir.

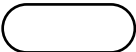

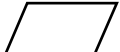
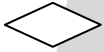








Blokların qrafik şəkildə ifadə olunması üçün Proqram Sənədlərinin Vahid Sistemi (PSVS) çərçivəsində standart qəbul olunmuşdur.

Alqoritmin blok-sxemini bütöv şəkildə qurmaq məsləhətdir, lakin lazım gəldikdə, blokları birləşdirən xətləri qırmaq mümkündür. Bu zaman həmin hissələr arasında əlaqə birləşdirici çevrə vasitəsilə göstərilir. Əgər blok-sxem bir neçə səhifədə yerləşirsə, blok-sxemin hissələri arasında əlaqələr səhifəarası birləşdirici vasitəsilə yaradılır.

Alqoritmin blok-sxemlə təsvirində hər bir mərhələnin məzmunu, mərhələlərin icra ardıcılığı, təkrarlanan hissələr (dövrələr) aydın görünür.

## Blokların şərti işarələri

- |                                  |   |  |
|----------------------------------|---|--|
| • Başlanğıc və ya son -          |  | Prosesin başlanmasını və ya sonunu bildirir  |
| • Proses (hesab) bloku -         |  | Verilənlərin qiymətini, təsvir formasını və ya yerləşməsinə dəyişən əməliyyat və ya əməliyyatlar qrupu |
| • Daxiletmə-xaricetməbloku -     |  | Verilənlərin kompyutərə daxil edilməsi və ya kompyuterdən xaric edilməsi                               |
| • Şərt (seçmə, məntiqi blok) -   |  | Müəyyən şərtə əsasən olaraq hesablama istiqamətinin seçilməsi  |
| • Dövr bloku -                   |  | Dövri strukturlu alqoritmlərin təsviri   |
| • Çapetmə -                      |  | Nəticələrin kağıza köçürülməsi   |
| • Maqnit disk -                  |  | Maqnit diskə daxil və ya xaric edilməsi  |
| • Altalqoritm və ya altproqram - |  | Əvvəlcədən hazırlanmış alqoritmlərdən (altalqoritm) və ya proqramlardan (altproqram) istifadə olunması |
| • Birləşdirici -                 |  | Səhifə daxilində bloklararası əlaqələrin göstəricisi   |
| • Səhifəarası birləşdirici -     |  | Müxtəlif səhifələrdə yerləşən alqoritm   |



- İzahat -



hissələri arasında əlaqələr  
Bloka izahat verilməsi

## Alqoritmın strukturu

İstənilən hesablama prosesi aşağıdakı tipik (elementar) alqoritmik strukturların kombinasiyasından təşkil olunur:

- Xətti
- Budaqlanan
- Dövri (təkrarlanan)

## Xətti alqoritmik struktur

Xətti alqoritmik struktur iki və daha çox prosesin ardıcılığından ibarət olur. Onun tərkibində şərt (seçmə) bloku olmur. Bu strukturu sxematik olaraq belə göstərmək olar:



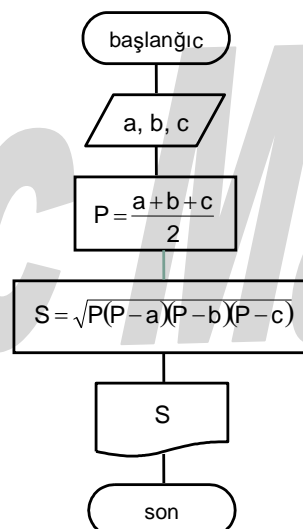
## Xətti alqoritmik struktura aid misal

Tərəfləri  $a$ ,  $b$ ,  $c$  olan üçbucağın sahəsinin hesablanması alqoritm.

Bilirik ki, üçbucağın sahəsi

$$S = \sqrt{P(P-a)(P-b)(P-c)}$$

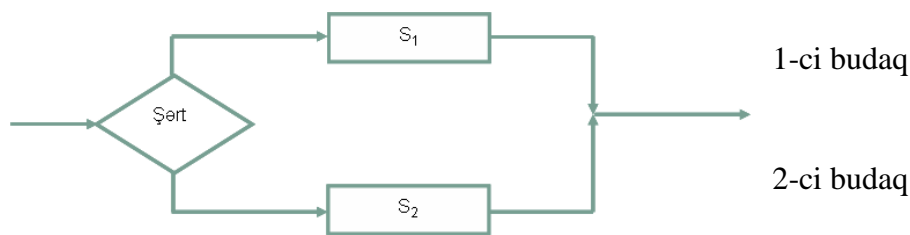
düsturu ilə hesablanır. Burada  $P = \frac{a+b+c}{2}$ .



## Budaqlanan alqoritmlər

Budaqlanan alqoritmik struktur – tərkibində məntiqi blok olan hesablama prosesini təsvir edir. Hər bir budaqlanma nöqtəsi uyğun məntiqi blokla təyin edilir. Bu blokda müəyyən kəmiyyətlərin (ilkin verilənlərin, aralıq nəticələrin və s.) bu və ya digər şərti ödəyib-ödəməməsi yoxlanılır və nəticədən asılı olaraq sonrakı gedişin istiqaməti seçilir.

Budaqlanan strukturu sxematik olaraq belə göstərmək olar:



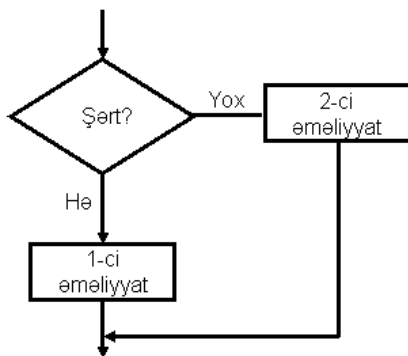
Bu halda nəzərdə tutulan **P** şərt ödənilirsə **S1** əməliyyatı, həmin şərt ödənilmərsə, **S2** əməliyyatı yerinə yetirilir. Beləliklə, alqoritmə budaqlanma baş verir.

## Budaqlanan alqoritmın növləri

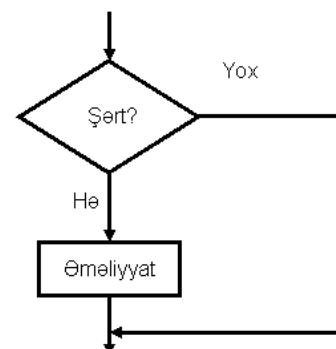
Blok-sxemdə hər hansı şərtədən asılı olaraq bütün hesablama istiqamətləri göstərməlidir. Lakin proqramın icrası zamanı bu istiqamətlərdən yalnız birinə görə hesablama aparılır. Seçilən hər bir budağa görə hesablama prosesi sonlu nəticəyə gətirib çıxarmalıdır.

Alqoritmik dildə budaqlanan struktur iki cür olur:

- tam formalı budaqlanma



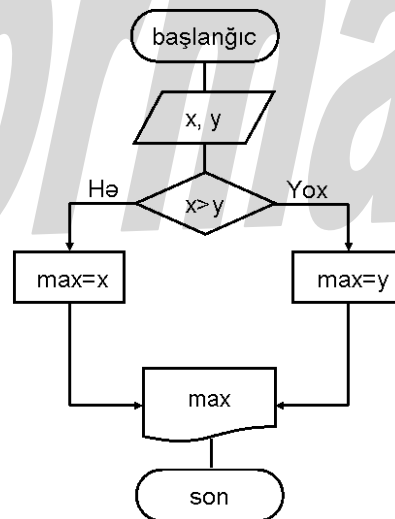
- natamam formalı budaqlanma





## Budaqlanan alqoritmik struktura aid misal

$x$  və  $y$  ədədlərindən ən böyüyünün tapılması alqoritmində aid misala baxaq. Tutaq ki, bu ədədlərdən maksimumunun tapılması məsələsinin alqoritm blok-sxem şəklində təsvir edilməsi tələb edilir. Burada  $X$ -in  $Y$ -dən böyük olması yoxlanılır və nəticədən asılı olaraq ya 1-ci, ya da 2-ci əməliyyat seçilir.

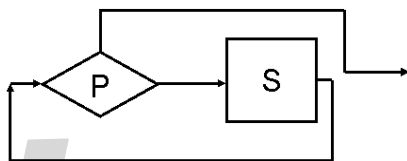


## Dövri alqoritmlər

Proqramlaşdırmada tez-tez eyni əməliyyatlar qrupunun çoxlu sayda təkrar olunması lazım gəlir. Bu halda dövr alqoritmindən istifadə olunur.

Hesablama prosesinin təkrarən yerinə yetirilən hissəsinə *dövr* deyilir.

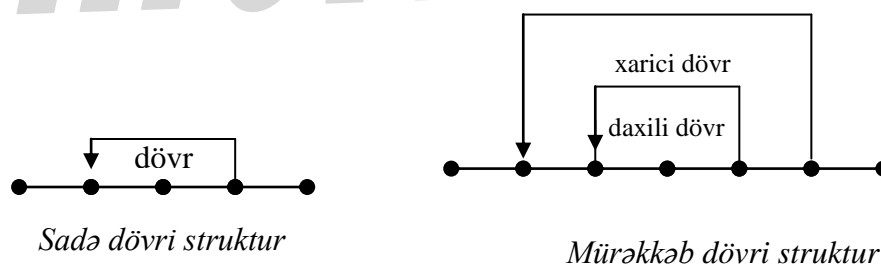
Aşağıdakı blok-sxemdə **S** operatoru **P** şərti ödəniləndiyi halda dəfələrlə yerinə yetirilir:



## Dövri alqoritmın növləri

Təcrübədə çox rast gəlinən dövrü hesablama proseslərində məsələnin( və ya onun bir hissəsinin) həlli eyni hesablama düsturları ilə dəyişənlərin müxtəlif qiymətləri üçün bir neçə dəfə təkrarən hesabların aparılmasını tələb edir. Hesablama prosesinin təkrarən yerinə yetirilən hissəsinə dövr deyilir. Təkrar yerinə yetirilən əməliyyatlar hissəsi dövrün gövdəsi adlanır. Amma "təkrar" sözü "sonsuzluq" ilə eyniləşdirilməməlidir. Alqoritmin sonsuzluğu onun nəticəvilik xassəsinin pozulmasıdır.

Dövrü strukturlar iki tip ola bilər: **sadə** və **mürəkkəb**. Sadə struktur bir, mürəkkəb struktur isə biri digərinə daxil olan iki və daha çox dövrədən ibarət olur. Bu strukturları sxematik olaraq belə göstərmək olar:



Proqramlaşdırma dillərində dövrü strukturların reallaşdırılması üçün **önşərtli** və **sonşərtli** dövrü operatorlardan istifadə olunur. Önşərtli operatorlarda dövrədən çıxış şərti dövrün əvvəlində, sonşərtli operatorlarda isə dövrün sonunda yoxlanılır. Önşərtli struktur WHILE ( nə qədər ki...) və FOR (üçün) operatoru ilə, sonşərtli struktur isə REPEAT ... UNTIL operatorları ilə yerinə yetirilir.

