

## 8- AMALIY MASHG'ULOT

### Mavzu: Iterativ va rekursiv algoritmlar.

**Ishning maqsadi:** Iterativ va rekursiv algoritmlar haqida tushuncha berish, hamda dasturlar tuzish ko'nikma va malakasini shakllantirish.

#### I. Nazariy qism

Boshlang'ich va oraliq ma'lumotlarni masalani yechish natijasiga aylantiradigan jarayonni bir qiymatli qilib, aniqlab beradigan qoidalarining biror bir chekli ketma-ketligi algoritm ekanligi yuqoridagi mavzulardan bizga ma'lum.

Buning mohiyati shundan iboratki, agar algoritm ishlab chiqilgan bo'lsa, uni yechilayotgan masala bilan tanish bo'lmagan biron bir ijrochiga, shu jumladan kompyuterga ham bajarish uchun topshirsa bo'ladi va u algoritmning qoidalariga aniq rioya qilib masalani yechadi. Quyida rekursiv algoritmgaga to'xtalamiz. Avvalo, rekursiv o'zi nima degan savolga javob beramiz. Agar obyektning tarkibiy qismlari obyektning o'zi orqali aniqlansa, u holda bu obyekt rekursiv deyiladi. Rekursiya nafaqat matematikada, balki kundalik hayotimizda ham ushlab turadi.

Rekursiya o'z kuchini ayniqsa, matematik ta'riflarda namoyon etadi. Eng tanish misollar sifatida natural sonlar, daraxtsimon tuzilmalar va ba'zi funktsiyalarni keltirishimiz mumkin:

1. Natural sonlar:

a) 0 natural son hisoblanadi.

b) Natural sondan keyin keluvchi son natural hisoblanadi.

2. Daraxtsimon tuzilmalar:

a)  $\emptyset$  daraxt hisoblanadi (va bo'sh daraxt deyiladi).

b) Agar  $t_1$  va  $t_2$  –daraxtlar bo'lsa,  $t_1$  va  $t_2$ ning avlodlaridan iborat tugundan tashkil topgan tuzilma ham daraxt deyiladi (ikkilik yoki binar daraxt).

3. Faktorial funktsiya  $f(n)$ :

$f(0)=1$

$n>0$  bo'lganda  $f(n)///$

Ko'rinib turibdiki, rekursiyaning kuchi cheksiz obyektlar to'plamini chekli mulohaza orqali aniqlash imkoniyatida namoyon bo'ladi. Xuddi shunday chekli sonli hisoblashlarni chekli dastur orqali tavsiflash mumkin. Bunda dastur oshkor tsikllarni o'z ichiga olmasligi ham mumkin. Ammo rekursiv algoritmlar, avvalo, yechilayotgan masala, hisoblanayotgan funktsiya yoki qayta ishlanayotgan ma'lumotlar tuzilmasi rekursiv ravishda berilgan holda o'rinalidir.

Umumiy holda, R rekursiv dastur (R ni o'z ishiga olmagan) S ko'rsatmalarning va R ni o'zining ketma-ketligidan iborat R birlashma (kompozitsiya) sifatida ifodalanishi mumkin:

$R=...$

Dasturlarning rekursiv ifodasi uchun zaruriy va yetarli vosita bo'lib protseduralar xizmat qiladi. Bu protseduralar ko'rsatmalar to'plamiga nom berish imkonini yaratib, bu nom orqali dastur bajarilish jarayonida ko'rsatmalar chaqirilishi mumkin.

Agar R protsedura oshkor holda o'ziga murojaatni o'z ichiga olsa, bunday protsedura oshkor rekursiv deyiladi. Agar R protsedura R ga to'g'ridan-to'g'ri yoki bilvosita murojaatni o'z ichiga olgan boshqa bir Q protsedurani tarkibiga olsa, u holda R bilvosita rekursiv deyiladi.

Rekursiv protseduralar cheksiz hisoblashlarga yo'l oshganligini hisobga olsak, uni to'xtatish muammosini ham hal etishimiz zarur. Fundamental talablar shundan iboratki, qaysidir vaqt momentiga kelib, bajarilmay qoladigan shunday V shart bajarilgandagina R rekursiv protseduraga murojaatlar amalga oshirilishi kerak.

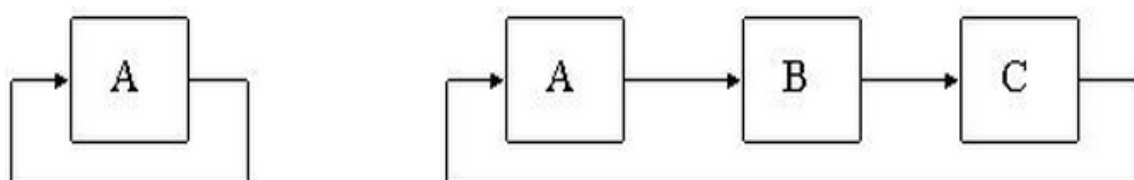
Shuning uchun ham rekursiv algoritmlarning sxemasini quyidagi ko'rinishlardan biri bilan ifodalash mumkin:

$P=IF\ B\ THEN\ P[S,P]\ END$

P=...

Ma'lumki, har qanday protsedura yoki funktsiya boshqa protsedura yoki funktsiyalarga murojaatni o'z ichiga olishi mumkin. Xususiyl holda, agar bu murojaat protsedura yoki funktsiyaning o'ziga murojaatdan iborat bo'lsa, bu protsedura yoki funktsiya rekursiv deyiladi.<sup>1</sup>

Umumiy holda, rekursiv protsedura va funktsiyalarni quyidagi sxema ko'rinishida ifodalashimiz mumkin:



Rekursiv protseduraga misol sifatida quyidagi qism dasturni tahlil etaylik:

```
procedure Res(a: integer);
```

```
begin
```

```
  if a>0 then
```

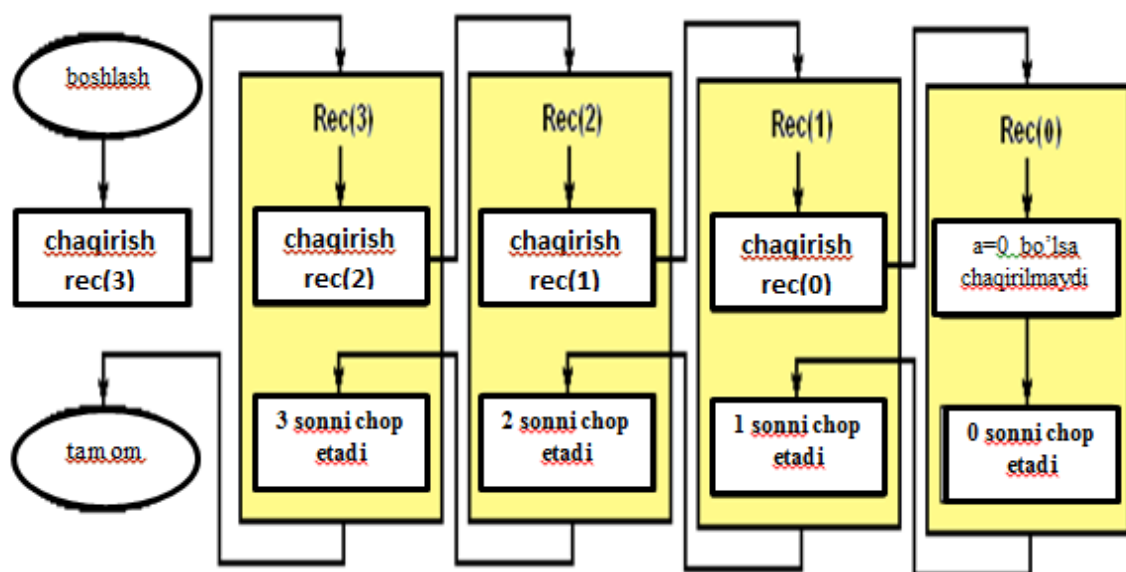
```
    Res(a-1);
```

```
    writeln(a);
```

```
end;
```

Tushunib olish uchun asosiy dasturdan Res(3) ko'rinishdagi murojaatda protsedura qanday ishlashini ko'rib chiqaylik.

Quyidagi blok-sxemada operatorlarning bajarilish ketma-ketligi tasvirlangan:



Boshlanish Res(3)ga murojaat Tugash

1-misol: Rekursiv protsedura ishini tasvirlovchi blok-sxema.

Res protsedurasi dastlab a=3 parametr bilan chaqiriladi, ya'ni Res(3) tarzidagi murojaat amalga oshiriladi. Uning tarkibida esa a=2 parametrli Res(2) tarzidagi murojaat mavjud. Hali dastlabki murojaat tugamasdan keyingi protseduraga murojaat tashkil etiladi, u tugamasa dastlabki Res(3) protsedura ishini tugatmaydi. Protседuraga murojaat a=0 ga davom etadi. Demak, bir vaqtning o'zida protsedura 4 marta chaqirilib ishlaydi. Bir vaqtda bajariladigan protseduralar soni rekursiya chuqurligi deyiladi.

<sup>1</sup> Н.Вирт. Алгоритмы и структуры данных. Москва – 2010 (132-136с)

To'rtinchi marta chaqirilgan Res(0) protsedura 0 sonini chop etadi va o'z ishini yakunlaydi. Shundan so'ng boshqaruv Res(1)ni chaqirgan protseduraga o'tadi va 1 ni chop etadi. Xuddi shu tariqa barcha protseduralar tugaguncha jarayon davom etadi. Natijada to'rtta 0, 1,2,3 sonlarini chop etiladi.

## II. Amaliy qism

**1-Misol.** Sonni ikkilik sanoq sistemasiga o'tkazish masalasini tsikl operatori va rekursiya orqali hal etishni qarab chiqamiz.

Ma'lumki, ikkilik sanoq sistemadagi sonlarni hosil qilish uchun berilgan sonni sanoq sistema asosi bo'lan 2ga bo'lish orqali amalga oshiriladi.

Agar berilgan son  $x$  bo'lsa, uning ikkilik sistemadagi ifodasida oxirgi raqam  $S1=x \bmod 2$

Ikkiga bo'lishda bo'linmaning butun qismini olamiz:

$$X2=x \div 2$$

Bu jarayon butun qism nolga teng bo'lguncha davom etadi. Bu amalga qoldiq 0 ga teng bo'lguncha davom etadi. Dastur rekursiyadan foydalanilgan holda quyidagi ko'rinishda bo'ladi:

**prosedure BinaryRepresentation(x: integer);**

**var**

**s, x: integer;**

**begin**

{ Birinchi chaqiruv. Protsedura murojaat qilinish tartibida amalga oshiriladi }

**s := x mod 2;**

**x := x div 2;**

{ Rekursiv murojaat }

**if x>0 then**

**BinaryRepresentation(x);**

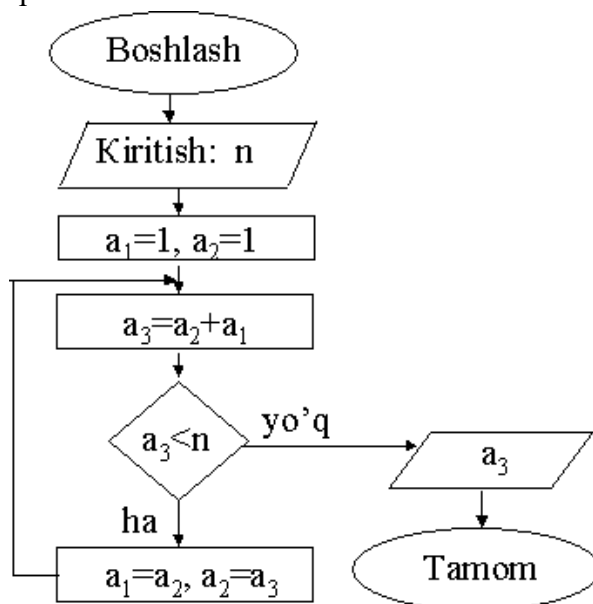
{ Ikkinchi blok. Teskari tartibda bajariladi }

**write(s);**

**end;**

Bunday algoritmgaga yana *misol* sifatida Fibonashchi sonlarini keltirish mumkin. Ma'lumki, Fibonashchi sonlari quyidagicha aniqlangan.

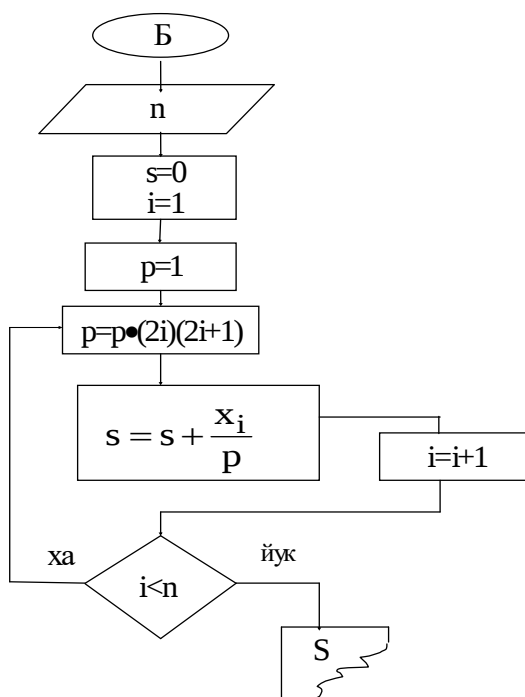
$a_0, a_1, a_2, a_3, \dots$  Bu rekkurent ifoda algoritmgaga mos keluvchi blok-sxema 2.15-rasmda keltirilgan. Eslatib o'tamiz formuladagi  $i$ -indeksga hojat yo'q, agar Fibonashchi sonining nomerini ham aniqlash zarur bo'lsa, birorta parametr-kalit kiritish kerak bo'ladi.



Fibonachchi sonlarining  $n$ - hadini hisoblash algoritmi.

$$\text{2-misol. } S = \sum_{i=1}^n \frac{x_i}{(2i + i)!}$$

Bu ifoda  $i$  ning har bir qiymatida faktorialni va yig'indini hisoblashni taqozo etadi. Shuning uchun avval faktorialni hisoblashni alohida ko'rib chiqamiz. Quyidagi rekkurent ifoda faktorialni kam amal sarflab qulay usulda hisoblash imkonini beradi:



R=1

R=R\*2i\*(2i+1)

Haqiqatan ham,  $i=1$  da  $3!$  ni,  $i=2$  da  $R=3!*4*5=5!$  ni va hakozi tarzda  $(2i+1)!$  ni yuqoridagi rekkurent formula yordamida hisoblash mumkin bo'ladi. Bu misolga mos keluvchi blok-sxema quyida keltirilgan.

### Rekursiv funksiya masalalari

1. [2] Fibonachchi  $f_n$  sonlarni topuvchi Rekursiv funksiya bo'lmagan  $\text{int } f(\text{int } n)$  yozing.

Fibonachchi sonlari  $f_1 = 1, f_2 = 1, f_n = f_{n-1} + f_{n-2}$ . munosabatlar orqali aniqlanadi.

2. [+2] Fibonachi  $f_n$  sonlarni topuvchi Rekursiv funksiya bo'lgan  $\text{int } f(\text{int } n)$  yozing. Fibonachchi sonlari  $f_1 = 1, f_2 = 1, f_n = f_{n-1} + f_{n-2}$ . munosabatlar orqali aniqlanadi

### Yechish:

Rekursiv funksiyalar

[+2] Fibonachchi  $f_n$  sonlarni topuvchi Rekursiv funksiya bo'lgan  $\text{int } f(\text{int } n)$  yozing

1 Fibonachchi sonlari  $f_1 = 1, f_2 = 1, f_n = f_{n-1} + f_{n-2}$ . munosabatlar orqali aniqlanadi.

$\text{int } f35(\text{int } x)$

{

counter++;

if(x<=1) return 1;

if(x==2) return 1;

return f35(x-1) + f35(x-2);

}

{

myofs << "masala 5. Рекурсия" << endl;

counter=0; int m3=f35(3);

myofs << "f35(3)=" << m3 << "co=" << counter << endl;

```

counter=0; int m4=f35(4);
myofs << "f35(4)=" << m4 <<"co="<<counter << endl;
counter=0; int m5=f35(5);
myofs << "f35(5)=" << m5 <<"co="<< counter << endl;
counter=0; int m6=f35(6);
myofs << "f35(6)=" << m6 <<"co="<< counter << endl;
counter=0; int m7=f35(7);
myofs << "f35(7)=" << m7 <<"co="<< counter << endl;
counter=0; int m10=f35(10);
myofs << "f35(10)=" << m10<<"co="<< counter << endl;
counter=0; int m20=f35(20);
myofs << "f35(20)=" << m20<<"co="<< counter << endl;
counter=0; int m30=f35(30);
myofs << "f35(30)=" << m30<<"co=" << counter << endl;
counter=0; int m35=f35(35);
myofs << "f35(35)=" << m35<<"co=" <<counter << endl;
}

```

Fayldagi natija:

Rekursiya

Значение f35(3) равно 2 counter= 3

Значение f35(4) равно 3 counter= 5

Значение f35(5) равно 5 counter= 9

Значение f35(6) равно 8 counter= 15

Значение f35(7) равно 13 counter= 25

Значение f35(10) равно 55 counter= 109

Значение f35(20) равно 6765 counter= 13529

Значение f35(30) равно 832040 counter= 1664079

Значение f35(35) равно 9227465 counter= 18454929

### III. MAVZU BO'YICHA TOPSHIRIQLAR.

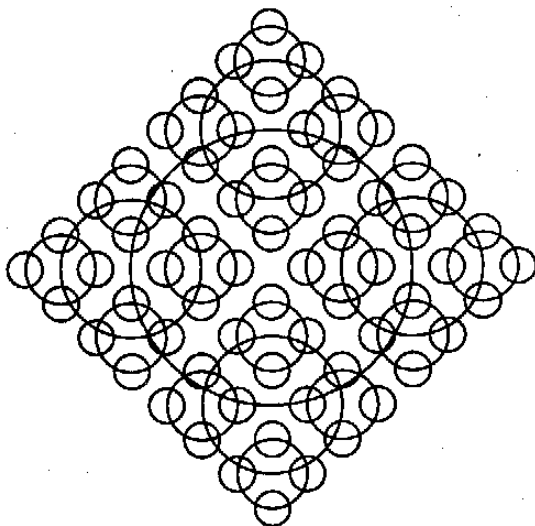
Bu bo'lim masalalarini yechishga o'tishdan oldin quyidagilarni eslash kerak:

O'z-o'zini chaqirib oluvchi protsedura va funktsiya rekursiv deyiladi;

jarayonni tugatish uchun rekursiv funktsiya (protsedura) algoritmda funktsiyaning (protseduraning) bevosita tugashini ta'minlovchi yo'nalish, albatta, bo'lishi kerak.

Faktorialni hisoblovchi funktsiyani va uning ishlashini tekshiruvchi dasturni yozing.

Quyida keltirilgan bezakni ekranga chiqaruvchi dasturni yozing.

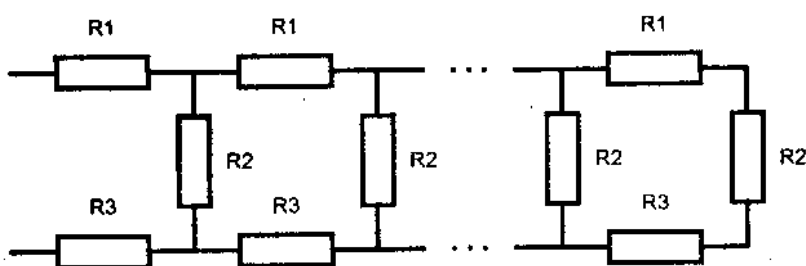


1. Sxemasi qarshiligini

miqdori va zanjir tartibi ( $R_2$  qarshiliklar soni) dastur ishlashi vaqtida kiritilishi kerak.

quyida keltirilgan elektr zanjir

hisoblash dasturini yozing. qarshiliklar



2. Yuqorida keltirilgan elektr zanjir sxemasini ekranga chizuvchi dasturni yozing. Zanjir tartibi dastur ishlash vaqtida kiritilishi kerak.

3. Yiqindini  $X$  ning haqiqiy qiymatlarida hisoblang.  $S = \sum_{n=1}^k \frac{X^{2n}}{(2n)}$

4. Yig'indini  $X$  ning haqiqiy qiymatlarida hisoblang.  $S = \sum_{n=1}^k \frac{X^{4n+1}}{4n+1}$

5. Yiqindini  $X$  ning haqiqiy qiymatlarida hisoblang.  $S = \sum_{n=1}^k (-1)^n \frac{\cos nx}{n^2}$

6. Yig'indini  $X$  ning haqiqiy qiymatlarida hisoblang.  $S = \sum_{n=1}^k \frac{n^2 + 1}{n!} \left( \frac{X}{2} \right)^n$

7.  $y=e^x$  funksiyaning qiymatini argumentning quyidagi o'zgarishlar sohasidagi  $1 \leq X \leq 2$  har bir nuqtasida qatorda yoyib ( $n=15$ ) hisoblang. qator yig'indisining har umumiy hadi:  $S = \sum_{n=1}^k \frac{X^n}{n!}$

8.  $S = \sum_{i=1}^n \frac{X_i}{(2i + i)!}$  ifodani hisobdash uchun rekkurent ifodasini keltiring.

9. Quyidagi misolga algoritm tuzing:  $S = \sum_{i=0}^3 (2i + 3)!$

Quyidagi misolga algoritm tuzing:  $Y = \sum_{i=1}^3 \frac{(2i + 2)!}{(3i + 4)!}$