

LABORATORIYA ISHI №11

MAVZU: REKURSIYA VA REKURSIV FUNKSIYALAR.

Ishning maqsadi: Talabalarda rekursiya va rekursiv funksiyalar bo'yicha ko'nikma va malakalarini shakllantirish.

Nazariy qism:

Rekursiya – matematika va kompyuter ilmda fundamental tushunchalardan biri hisoblanadi. Muayyan sinf masalalari uchun samarali dasturlash usullaridan biri rekursiya hisoblanadi. Uning yordamida biz murakkab sonli tahlil, kombinatorika, algoritmlarni o'girish, ro'yxat tuzilmalari bilan ishlash va boshqa masalalarni hal qilishimiz mumkin. Bunday hollarda, algoritmlar (dasturlar) iteratsiya bilan taqqoslaganda kichik hajmda va kompyuterda bajarilishida kamroq vaqt sarflanadi.

Rekursiya deganda – biz odatda o'ziga-o'zi murojaat qiladigan funktsiyani tushunamiz. Masalan, $n! = (n - 1)! * n$

Dasturlashda rekursiv protsedura (funksiya) protsedurani (funtsiyani) o'ziga chaqirish usuli sifatida tushuniladi. Iteratsiya deganda – qandaydir amalning ko'p marta takrorlanish natijasi tushuniladi. Masalan, $n! = 1 * 2 * 3 * 4 * ... * n$

Masalalarning keng sinflari orasida rekursiv protseduralar (funktsiyalar) yordamida bir xil turdagi, lekin kichikroq o'lchamdagi masalalarga keltirilishi mumkin bo'lgan masalalarni tasvirlash qulay.

Rekursiyani tahlil qilishning umumiy metodologiyasi uch bosqichni o'z ichiga oladi:

1. Yechimga bog'liq bo'lgan turli elementlarni, xususan, yechilayotgan muammoning o'lchamini aniqlashdan iborat bo'lgan masalani parametrlashtirish. Har bir rekursiv murojaatdan keyin o'lcham kamayishi kerak.

2. Trivial holat va uning yechimini izlash. Qoidaga ko'ra, bu rekursiyaning asosiy bosqichidir va masalaning o'lchami ko'pincha 0 yoki 1 ni tashkil qiladi.

3. Masalani bir xil turdagi, lekin o'lchami kichik bo'lgan bir yoki bir nechta masalalarga qisqartirishni maqsad qilgan umumiy holatning dekompozitsiyasi.

Rekursiya. Dasturlash tillarida rekursiv dastur o'z-o'ziga murojaat qiluvchi

dasturdir (matematikada rekursiv funksiya shu funksiyaning o'zi yordamida aniqlanadigan funksiyadir). Rekursiv dastur o'ziga cheksiz murojaat qila olmaydi, shuning uchun rekursiv dasturning ikkinchi muhim xususiyati dasturni chaqirishni to'xtatish imkonini beruvchi shartning mavjudligidir. Shunday qilib, dasturlashda rekursiya masalani bir xil masalaga qisqartirish, ammo oddiy ma'lumotlarni manipulyatsiya qilish sifatida aniqlash mumkin.

Rekursiyami yoki iteratsiya?

– Rekursiv usul xotira stackidan foydalangani uchun iterativ usuldan ko'ra xotiradan qo'shimcha joy oladi. Ya'ni har bir chaqirilgan funksiya call stackda saqlanishi uchun operativ xotiradan joy kerak bo'ladi. Iterativ usulda esa bunday muammo deyarli yo'q.

– Aynan shu call stackga funksiyalar qo'shilishi va ularni qayta chaqirib olish rekursiya ishini iteratsiyaga qaraganda sekinlashtiradi.

– Ba'zi holatlarda rekursiv yechim ancha soddaroq. Ayniqsa, ba'zi masalalarning iterativ shakli juda ham uzun bo'lib ketishi mumkin. Rekursiya esa kodni bir necha barobar qisqartirib berishi mumkin.

– Aksariyat tuzilmalar va algoritmlarni rekursiyasiz tasavvur qilib bo'lmaydi. Tree, Graph, Heap, QuickSort, MergeSort, ... Bu ro'yxatni juda uzoq davom ettirish mumkin. Ayniqsa, murakkab tuzilmalar bo'lgan Tree va Graphlarda rekursiya har qadamda uchraydi. Dasturlashni esa ularsiz tasavvur qilib bo'lmaydi, bu esa o'z o'rnida rekursiya qanchalik muhimligini belgilab beradi

LABORATORIYA ISHINI BAJARISH UCHUN NAMUNA:

1-misol. Rekursiv funksiyadan foydalanib, Fibonachi ketma-ketligi N sonni toping. Fibonachchi ketma-ketligi quyidagi sonlar 0 1 1 2 3 5 8 13... ketma-ketligi orqali hosil qilinadi, ya'ni ketma-ketlidagi had o'zidan oldingi ikkita hadning yig'indisiga teng. $F(N)$ yordanida Fibonachi qatorining N – chi hadini belgilaymiz va uni quyidagi rekursiv funksiya orqali yozish mumkin.

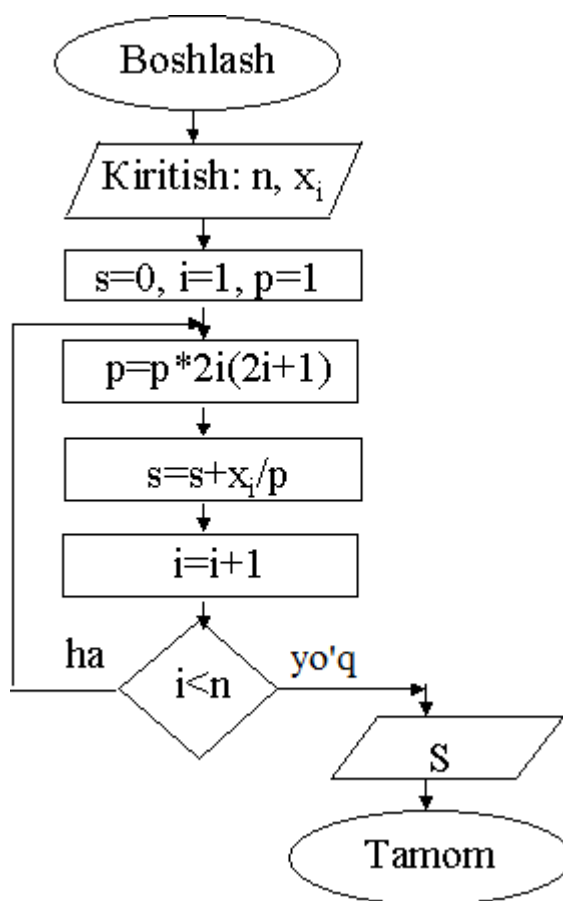
$$\begin{aligned} F(N) &= F(N-1) + F(N-2), \quad N \geq 3 \\ F(1) &= 1 \text{ va } F(2) = 1 \end{aligned} .$$

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

Bunda i indeksning har bir qiymatida faktorial va yig'indini hisoblash taqozo etiladi. Shuning uchun avval faktorialni hisoblashni alohida ko'rib chiqamiz. Quyidagi rekkurent ifoda faktorialni kam amal sarflab qulay usulda hisoblash imkonini beradi:

$$(1) R=1; (2) R=R*2i*(2i+1).$$

Haqiqatan ham, $i=1$ da $3!$ ni, $i=2$ da $R=3!*4*5=5!$ ni va hokazo tarzda $(2i+1)!$ ni yuqoridagi rekkurent formula yordamida hisoblash mumkin bo'ladi. Bu misolga mos keluvchi blok-sxemasi 11.1-rasmda keltirilgan.



11.1-rasm. Rekurrent algoritmgaga doir blok-sxema.

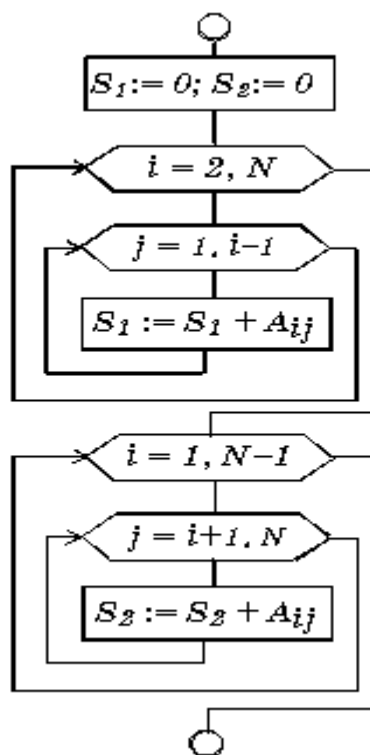
3-misol. $A(N, N)$ massivdagi bosh digonaldan yuqorida joylashgan elementlar va pastda joylashgan elementlar yig'indilarini hisoblang.

Berilgan	Natija
----------	--------

$N=3$	$\begin{pmatrix} 1 & 2 & 4 \\ 3 & 1 & 3 \\ 2 & 1 & 2 \end{pmatrix}$	$S_1=6$ $S_2=9$
-------	---	--------------------

Algoritmi	Algoritmning bajarilishi																																
alg Yig' (but N, haq jad A[1:N, 1:N], haq S1, S2) arg N,A natija S1,S2 boshl but i, j S1:=0; S2:=0 sb i uchun 2 dan N gacha sb j uchun 1 dan i-1 gacha S1:=S1 + A[i, j] so so sb i uchun 1 dan N-1 gacha sb j uchun i+1 dan N gacha S2:=S2 + A[i, j] so so tamom	<table><tr><th>i</th><th>j</th><th>S1</th><th>S2</th></tr><tr><td></td><td></td><td>0</td><td>0</td></tr><tr><td>2</td><td>1</td><td>0+3=3</td><td></td></tr><tr><td>3</td><td>1</td><td>3+2=5</td><td></td></tr><tr><td></td><td>2</td><td>5+1=6</td><td></td></tr><tr><td>1</td><td>2</td><td></td><td>0+2=2</td></tr><tr><td></td><td>3</td><td></td><td>2+4=6</td></tr><tr><td>2</td><td>3</td><td></td><td>6+3=9</td></tr></table>	i	j	S1	S2			0	0	2	1	0+3=3		3	1	3+2=5			2	5+1=6		1	2		0+2=2		3		2+4=6	2	3		6+3=9
i	j	S1	S2																														
		0	0																														
2	1	0+3=3																															
3	1	3+2=5																															
	2	5+1=6																															
1	2		0+2=2																														
	3		2+4=6																														
2	3		6+3=9																														

Blok-sxemasi fragmenti:



LABORATORIYA ISHINI BAJARISH UCHUN TOPSHIRIQLAR:

1. Qaysi masalalarda tarkibiy algoritmdan foydalaniladi?

2. Rekursiyaga oid algoritmlar nimadan iborat?

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

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

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

6. $S = \sum_{i=0}^3 (2i + 3)!$ hisoblansin.

7. $Y = \sum_{i=1}^3 \frac{(2i + 2)!}{(3i + 4)!}$ hisoblansin.

8. Ichma-ich joylashgan tsiklik algoritmlar, rekursiya va rekurrent algoritmlarga doir ma'lumotlar tayyorlang.

9. $N!$ - qiymatini hisoblovchi rekursiv funksiyani tuzing.

10. Quyidagilarni hisoblovchi rekursiv funksiyani tuzing:

a) Natural son raqamlari yig'indisi;

b) Natural son raqamlari miqdori.

11. Arifmetik progressiyaning birinchi hadi va ayirmasi berilgan bo'lsa rekursiv funksiya yordamida quyidagilarni aniqlang:

a) progressiyaning n -chi hadi;

b) progressiyaning dastlabki n - hadi yig'indisi.

12. Geometrik progressiyaning birinchi hadi va ayirmasi berilgan bo'lsa rekursiv funksiya yordamida quyidagilarni aniqlang:

a) progressiyaning n - hadi;

b) progressiyaning dastlabki n -hadi yig'indisi;

13. Berilgan $A(N)$ massivning komponentlarini o'ngga ikki pozisiya siklik siljishni amalga oshiring, ya'ni $A = (a_{N-1}, a_N, a_1, a_2, \dots, a_{N-2})$ vektorni hosil qiling.

14. $N \times M$ tartibli matritsa berilgan. Shunday B massiv tuzingki bunda agar matritsaning k -ustun elementlari simmetrik bo'lsa, uning k -elementiga 1, aks holda 0 qiymat bering.

15. $N \times M$ tartibli matritsa berilgan. Matritsaning "maxsus" elementlari soni k – ni aniqlang, "maxsus" element hisoblanadi, agar u o'z usunidagi boshqa qolgan elementlari yig'indisidan katta bo'lsa.

