

CHIZIQLI VA NOCHIZIQLI MA'LUMOTLAR TUZILMALARI

Tojimamatov Israiljon Nurmamatovich

Farg'ona davlat universiteti,

israiltojimatov@gmail.com

Ortiqaliyeva Hulkaroy Azizbek qizi

Farg'ona davlat universiteti talabasi,

soliyevmutavakil@gmail.com

Annatsiya: Chiziqli ma'lumotlar tuzilmasi - bu ma'lumotlarni tartibga solish va boshqarish uchun ishlatiladigan asosiy hisoblash tuzilmalaridan biridir. Chiziqli va nochiziqli ma'lumotlar tuzilmalari dasturlashda asosiy rol o'ynaydi. Shu boisdan bu mavzu muhim hisoblanadi.

Kalit so'zlar: Chiziqli tuzilmalar, nochiziqli tuzilmalar, massiv, bog'langan ro'yhat, navbat, stek, hash jadvallar, daraxtlar, graflar.

Аннотация: Линейная структура данных является одной из основных вычислительных структур, используемых для организации данных и управления ими. Линейные и нелинейные структуры данных играют ключевую роль в программировании. Поэтому эта тема важна.

Ключевые слова: Линейные структуры, нелинейные структуры, массив, связный список, очередь, стек, хеш-таблицы, деревья, графы.

Abstract: A linear data structure is one of the basic computational structures used to organize and manage data. Linear and non-linear data structures play a key role in programming. Therefore, this topic is important.

Key words: Linear structures, non-linear structures, array, linked list, queue, stack, hash tables, trees, graphs.

KIRISH

Barcha dastur yoki dasturiy mahsulotning asosida ikkita birlik yotadi: ma'lumotlar va ular ustida qandaydir amallar bajaradigan algoritmlar. Algoritmlar ma'lumotlarni biz yoki dastur uchun foydali bo'lgan axborot ko'rinishiga keltirib beradi. Algoritmlar shu ma'lumotlar ustida amallarni (o'qish, yozish, yangilash, o'chirish) samarali va tez bajara olishi uchun biz shu ma'lumotlarni ma'lum bir strukturaga solgan holda saqlashimiz kerak bo'ladi.

ADABIYOTLAR TAHLILI VA METODLARI

Ma'lumotlar tuzilmasi - bu ma'lumotlarni samarali o'qish va o'zgartirish imkonini beruvchi, ma'lumotlarni saqlash va boshqarishning bir formatga solingan shaklidir. Sodaroq qilib aytganda, ma'lumotlar tuzilmasi — bu ma'lumotlarning ma'lum bir strukturaga solingan, ular o'rtasida ma'lum bir bog'lanishlar yaratilgan

va ular ustida ma'lum amallar bajaruvchi funksiyalardan tashkil topgan birlikdir. Eng sodda ma'lumotlar tuzilmasiga misol qilib massivni ko'rsatishimiz mumkin.

Dasturlashda eng ko'p qo'llaniladiga ma'lumotlar tuzilmalariga quyidagilar kiradi:

1. Massiv (Array)
2. Bog'langan ro'yhat (Linked List)
3. Navbat (Queue)
4. Stek (Stack)
5. Hash jadvallar (Hash tables)
6. Daraxtlar (Trees)
7. Graflar (Graph)

Bundan tashqari barcha tuzilmalarning hamma joyda ingliz tilidagi nomi ishlatiladi va ularning nomi odatda tarjima qilinmaydi. Shu bois ingliz tilidagi nomlaridan foydalanamiz. Ma'lumotlar tuzilmalari ustida asosiy bajarilishi mumkin bo'lgan amallar Bularga ma'lumotlarni

- Ko'rib chiqish (Traversing)
- O'qib olish (Retrieving)
- Kiritish (Insertion)
- O'chirish (Deletion)
- Qidirish (Search)
- Saralash (Sorting)

Birlashtirish (Merging)lar kiradi. Turli xildagi ma'lumotlar tuzilmalarida esa yuqoridagi amallar turlicha tezlikda amalga oshadi. Masalan oddiyroq misol olaylik, array uchun ma'lumotni o'qib olish uchun $O(1)$ vaqt sarflansa, uni kiritish yoki o'chirish uchun $O(n)$ vaqt sarflanadi. Linked listda esa bular aksincha. Shuning uchun, masalan, sizning dasturingizda ma'lumotlar ko'p kiritilib, o'chirilsayu lekin kam o'qilsa, bunda ma'lumotlarni saqlashda arraydan foydalangandan ko'ra linked list qulayroq hisoblanadi. Lekin, ko'pincha holatda bir necha ma'lumotlar tuzilmalarini o'zini birlashtirgan gibrid ma'lumotlar tuzilmalaridan ham foydalaniladi.

NATIJALAR

Bog'langan ro'yxat (linked list) ma'lumotlar tuzilmasining bir turi bo'lib, unda elementlar ketma-ket joylashgan va har bir element boshqa bir elementga (yoki hech narsaga) bog'langan holda saqlanadi. Bu turdagi tuzilma dinamik ma'lumotlar tuzilmalaridan biri hisoblanadi va ko'pincha massivlarga alternativa sifatida ishlatiladi. Har bir element (tugun) faqat keyingi elementga bog'langan bo'ladi. Har bir tugun ikkita qismdan iborat. Ma'lumot qismi: tugunda saqlanadigan qiymat va bog'lanish qismi: keyingi tugunga ko'rsatkich. Har bir tugun oldingi va keyingi tugunlarga bog'langan. Har bir tugun uch qismdan iborat. Ma'lumot qismi, oldingi tugunga ko'rsatkich hamda keyingi tugunga ko'rsatkich. Bog'langan ro'yxat dinamik ravishda kengayishi yoki qisqarishi mumkin. Qo'shish va o'chirish operatsiyalari massivlarga qaraganda osonroq. Ma'lum bir indeksdagi elementga kirish uchun butun ro'yxatni

ko'rib chiqish kerak bo'lishi mumkin. Har bir tugun uchun qo'shimcha xotira (ko'rsatkichlar) talab qilinadi.

Stek (Stack) — bu ma'lumotlar tuzilmasining bir turi bo'lib, unda elementlar oxirgi kirgan birinchi chiqadi (Last In, First Out — LIFO) tamoyiliga asoslangan holda boshqariladi. Bu shuni anglatadiki, stekga qo'shilgan eng oxirgi element birinchi bo'lib olib tashlanadi. Yangi elementni esa stekning yuqori qismiga qo'shadi. Vazifalari: Eng yuqoridagi elementni olib tashlaydi va qaytaradi, stekning yuqori qismidagi elementni olib tashlamasdan ko'rishga imkon beradi, stek bo'sh yoki yo'qligini tekshiradi, so'nggi kirgan element birinchi bo'lib qaytariladi, tek elementlari vertikal tarzda "chiziqli" tartibda saqlanadi va ma'lumotlarni faqat stekning yuqori qismidan boshqarish mumkin. Elementlarga tasodifiy kirishning iloji yo'q.

Hash jadvallari (Hash Tables) — ma'lumotlarni saqlash va ularga tezkor kirish uchun ishlatiladigan ma'lumotlar tuzilmasidir. U ma'lumotlarni kalit-qiymat (key-value) juftlari shaklida tashkil qiladi. Hash jadvallari ko'pincha dictionary yoki map deb ham ataladi. Hash jadvali ma'lum bir kalitga mos keladigan qiymatni qidirish uchun hash funksiyasi deb ataluvchi matematik funksiyadan foydalanadi. Bu funksiya kalitni indeksga o'zgartiradi, so'ngra shu indeksdan foydalanib qiymatga murojaat qiladi. Kiruvchi kalitni jadvalning indeksiga aylantiradi. Vazifalari: kalitga mos qiymat hash jadvalida olingan indeksda saqlanadi, kalit uchun hash funksiyasi qayta chaqiriladi va tegishli indeksdan qiymat olinadi, hash funksiyasi bir nechta kalitga bir xil indeksni qaytarishi mumkin va bu holatni hal qilish uchun maxsus usullar qo'llaniladi. Afzalliklariga qidirish, qo'shish va o'chirish operatsiyalari o'rtacha holda $O(1)O(1)O(1)$ vaqt sarflashi, ma'lumotlarni kalit orqali boshqarish mumkinligi, dasturlashda ko'pincha ma'lumotlarni indekslash uchun ishlatilishi kiradi. Kamchiliklariga esa agar ko'plab kalitlar bir xil indeksni hosil qilsa, hash jadvalining samaradorligi pasayishi, yaxshi hash funksiyani topish muhimligi, hash jadvali ishlatilmagan bo'sh joylarni ham band qilishi kiradi.

Daraxtlar (Trees) — bu ma'lumotlar tuzilmasining ierarxik shakli bo'lib, unda elementlar tugunlar (nodes) deb ataladi va ular ota (parent) va farzand (child) munosabatlariga ega bo'ladi. Daraxt ma'lumotlar tuzilmasi tabiatdagi daraxtlarning teskari shaklini eslatadi: ildiz (root) yuqorida, shoxlar esa pastga tomon tarqaladi.

Graf (Graph) — bu ma'lumotlar tuzilmasi bo'lib, u tugunlar (vertices) va ularga bog'langan qirralar (edges) to'plamidan iborat. Graflar turli xil obyektlar o'rtasidagi munosabatlarni modellash va tahlil qilishda ishlatiladi. Uning xususiyatlariga quyidagilarni kiritish mumkin. Har bir tugun ma'lum bir obyektni ifodalashi, tugunlar o'rtasidagi bog'lanish yoki aloqalarni ifodalashi, qirralarga bog'langan qiymatlar (masofa, narx, vaqt va boshqalar) mavjudligi va bir tugundan boshqasiga boradigan tugunlar va qirralar ketma-ketligi bo'lishi.

XULOSA. Dasturlashda samarali yechimlarni ishlab chiqish uchun ma'lumotlar tuzilmalari va ularning ishlash tamoyillarini tushunish muhimdir. To'g'ri tanlangan ma'lumotlar tuzilmasi nafaqat kodni samarali qiladi, balki muammolarni hal qilishni ancha osonlashtiradi. Ma'lumotlar tuzilmalari samaradorlik va murakkablik o'rtasidagi muvozanatni ta'minlashga yordam beradi. Ularni yaxshi tushungan dasturchi har qanday dasturiy muammoni osonroq va tezroq hal qila oladi. Shuning uchun, ma'lumotlar tuzilmalari har bir dasturchining bilim asosini tashkil etadi. Ma'lumotlar tuzilmalari dasturlashning muhim poydevorini tashkil etadi. Ular ma'lumotlarni samarali boshqarish, saqlash va ularga ishlov berish imkoniyatini yaratadi. Har bir tuzilma muayyan muammolarni hal qilish uchun mo'ljallangan va to'g'ri tanlov dastur samaradorligini sezilarli darajada oshiradi. Ma'lumotlar tuzilmalari algoritmlarning ishlash tezligini belgilaydi va kodni optimallashtirishda hal qiluvchi rol o'ynaydi. Shuning uchun ularni o'zlashtirish har bir dasturchining muhim vazifasidir.

FOYDALANILGAN ADABIYOTLAR:

1.Tojimamatov, I., & Xurshidbek, R. (2023). KATTA HAJMLI MALUMOTLARNI QAYTA ISHLASHDA QOLLANILAYOTGAN TEXNOLOGIYALAR: NOSQL, MAPREDUCE, HADOOP, ERP, SAP NOSQL TEXNOLOGIYASI. ОБРАЗОВАНИЕ НАУКА И ИННОВАЦИОННЫЕ ИДЕИ В МИРЕ, 18(6), 54-60.

2.Kimyonazarova, D., Ne'matjonova, D., Ergasheva, B., & Tojimamatov, I. (2023, March). KATTA MA'LUMOTLAR BILAN ISHLASHDA HADOOP ARXITEKTURASI. In Международная конференция академических наук (Vol. 2, No. 3, pp. 96-99).

3. Tojimamatov, I., & Doniyorbek, A. (2023). KATTA HAJMLI MA'LUMOTLAR AFZALLIKLARI VA KAMCHILIKLARI. ОБРАЗОВАНИЕ НАУКА И ИННОВАЦИОННЫЕ ИДЕИ В МИРЕ, 18(6), 66-70.

4. Abdulxadov, N., Saminjonov, S., & Tojimamatov, I. (2023). MA'LUMOTLAR VA AXBOROTLARNI VIZUALIZATSIYA QILISH USULLARI, INTERAKTIV MEKANIZMLAR. Евразийский журнал технологий и инноваций, 1(4), 7-18.

5.Qodirjonova, N., Tursunova, N., Parpiboyev, N., & Tojimamatov, I. (2023). BIR KOMPYUTERDA KATTA MA'LUMOTLAR BILAN ISHLASH. Центральноеазиатский журнал образования и инноваций, 2(4), 104-111.