

KONTEYNERLAR BILAN ISHLAYDIGAN ALGORITMLAR

Toshkent Davlat Iqtisodiyot Universiteti Andijon Fakulteti 2- bosqich AATT
yo'nalish 70/23 guruh talabalari

Abdumutaliyev Azamat

Nemadjonov Feruzbek

Ilmiy rahbar **Ismailov Alisher**

Anotatsiya. Ushbu maqolada konteynerlar bilan ishlaydigan algoritmlar va ularning dasturlash jarayonidagi o'rnini keng ko'lamda yoritiladi. Konteynerlar, odatda, ma'lumotlarni boshqarish, saqlash va ularga samarali ishlov berish uchun ishlatiladigan muhim ma'lumot tuzilmalari hisoblanadi. Konteynerlar yordamida turli xil ma'lumot turlarini tartiblash, qidirish, qo'shish va olib tashlash kabi amallarni tezkorlik va qulaylik bilan bajarish mumkin. Zamonaviy dasturlash tillarida, jumladan, C++ dasturlash tilida STL (Standard Template Library) kutubxonasi orqali turli xil tayyor konteynerlar taqdim etilgan bo'lib, ular yordamida murakkab ma'lumot tuzilmalari va ularning ustida bajariladigan algoritmlar soddalashtiriladi.

Konteynerlar bilan ishlaydigan algoritmlar dasturiy ta'minotning samaradorligini oshirishda katta ahamiyatga ega, chunki ular ma'lumotlarga oson va tezkor kirishni ta'minlaydi. Vector, Stack, Queue, Set va Map kabi konteynerlar turli xil ma'lumotlar bilan ishlashda yordam beradi va ular uchun mos algoritmlar ishlab chiqilgan. Misol uchun, Vector konteynerida tartiblash algoritmlari (Quick Sort, Bubble Sort) va qidirish algoritmlari (Linear Search, Binary Search) qo'llaniladi, Stack va Queue konteynerlarida esa push, pop, enqueue, va dequeue amallari bajariladi. Set konteyneri ma'lumotlarning noyobligini saqlashda samarali bo'lsa, Map konteyneri kalit-qiyamat juftligini boshqarishda qo'llaniladi.

Ushbu maqolada konteynerlar bilan ishlashning nazariy jihatlari bilan birga, amaliy misollar ham keltiriladi. C++ dasturlash tilida yozilgan kod misollarida konteynerlar bilan ishlaydigan algoritmlarning ishlash jarayoni aniq ko'rsatiladi. Bu esa dasturchilarga konteynerlar va ularning ustida bajariladigan algoritmlarni samarali ishlatishni o'rgatadi. Algoritmlarning samaradorligi va mosligini tushunish, dasturiy ta'minotning optimallashtirilishiga yordam beradi.

Maqolada konteynerlar bilan ishlashning asosiy tamoyillari, ularning afzalliklari va har bir turdagi konteynerga mos algoritmlarning samaradorligi batafsil yoritiladi.

Shu bilan birga, dasturlash amaliyotida konteynerlar bilan ishlashning qanday afzalliklari borligi ham tahlil qilinadi.

Kalit so'zlar: Konteyner, algoritm, ma'lumot tuzilmalari, C++, tartiblash, qidirish, vector, stack, queue, STL.

Annotation. Algorithms that work with containers and their role in the programming process are covered in this article. Containers are typically important data structures used to manage, store, and efficiently process data. Sorting, searching, adding and removing various types of information can be done quickly and conveniently with the help of containers. In modern programming languages, including the C++ programming language, various ready-made containers are provided through the STL (Standard Template Library) library, which simplify complex data structures and algorithms executed on them.

Algorithms that work with containers are of great importance in improving the efficiency of software, because they provide easy and fast access to data. Containers such as Vector, Stack, Queue, Set and Map help in working with various data and suitable algorithms have been developed for them. For example, sorting algorithms (Quick Sort, Bubble Sort) and search algorithms (Linear Search, Binary Search) are used in the Vector container, and push, pop, enqueue, and dequeue operations are performed in the Stack and Queue containers. While the Set container is effective in maintaining the uniqueness of data, the Map container is used to manage key-value pairs.

In this article, along with the theoretical aspects of working with containers, practical examples are presented. Code examples written in the C++ programming language clearly show the working process of algorithms working with containers. It teaches developers how to effectively use containers and the algorithms that run on them. Understanding the efficiency and suitability of algorithms helps in software optimization.

The article details the basic principles of working with containers, their advantages, and the effectiveness of algorithms suitable for each type of container. At the same time, the advantages of working with containers in programming practice are also analyzed.

Keywords: Container, algorithm, data structures, C++, sort, search, vector, stack, queue, STL.

Kirish. Dasturlashning muhim jihatlaridan biri bu ma'lumotlar bilan samarali ishlashdir. Ma'lumotlar strukturasi to'g'ri tanlash va ularni boshqarish

algoritmalarini o'rganish dasturchilarga yuqori samaradorlikka ega dasturlar yozishga yordam beradi. Dasturlashda turli xil ma'lumotlarni boshqarish uchun ishlatiladigan asosiy vositalardan biri konteynerlardir. Konteynerlar — bu turli xil ma'lumotlarni bir joyda saqlash va ularga turli amallarni bajarish imkoniyatini beruvchi ma'lumot tuzilmalari. Ular dasturchilar uchun ma'lumotlar ustida tezkor va oson ishlash imkonini yaratadi.

C++ dasturlash tilida konteynerlar bilan ishlashning asosiy vositasi bu Standard Template Library (STL) hisoblanadi. STL, turli xil ma'lumotlarni boshqarish uchun mo'ljallangan bir qator tayyor ma'lumot tuzilmalarini va algoritmalarini taqdim etadi. Ushbu kutubxona yordamida dasturchi turli xil konteynerlar (masalan, vector, stack, queue, set, map) va ularga mos algoritmalarini osongina ishlatishi mumkin. Buning natijasida dasturiy ta'minotni ishlab chiqish jarayoni sezilarli darajada tezlashadi va kod yanada toza va o'qiladigan bo'ladi.

Konteynerlar bilan ishlashda asosiy maqsad ma'lumotlarni samarali saqlash va ularga ishlov berishdan iborat. Har bir konteynerning o'ziga xos xususiyatlari va ularga mos keladigan algoritmalar mavjud. Masalan, vector — bu eng ko'p ishlatiladigan konteynerlardan biri bo'lib, u dinamik uzunlikka ega ketma-ket ma'lumot tuzilmasidir. Vector ichidagi elementlarga indeks orqali kirish mumkin va bu uni boshqa konteynerlarga qaraganda tezroq ishlashga olib keladi. Vector bilan ishlaydigan algoritmalar orasida tartiblash (masalan, Bubble Sort, Quick Sort) va qidirish (Linear Search, Binary Search) algoritmalar mavjud. Ushbu algoritmalar vector ichidagi elementlar bilan samarali ishlash uchun mo'ljallangan.

Yana bir muhim konteyner — bu stack (to'plar). Stack ma'lumotlarni saqlashda LIFO (Last In, First Out) tamoyiliga asoslanadi, ya'ni oxirgi kiritilgan element birinchi bo'lib chiqariladi. Stack dasturlashda ko'pincha rekursiv muammolarni hal qilishda yoki funksiyalarni chaqirish jarayonlarini boshqarishda ishlatiladi. Stack bilan ishlaydigan asosiy operatsiyalar — bu push (elementni qo'shish) va pop (elementni olish). Bu operatsiyalar o'zaro bog'liq bo'lib, stack ichidagi ma'lumotlarga to'g'ridan-to'g'ri kirish o'rniga, ularning ketma-ketligini boshqarishga imkon beradi.

Queue (navbat) esa FIFO (First In, First Out) tamoyiliga asoslanadi, ya'ni birinchi kiritilgan element birinchi bo'lib chiqariladi. Navbatlar odatda resurslarni boshqarish, vazifalarni ketma-ket bajarish yoki tarmoq orqali ma'lumot uzatish jarayonlarida qo'llaniladi. Navbat bilan ishlashda asosiy operatsiyalar — bu enqueue (elementni navbatga qo'shish) va dequeue (elementni navbatdan chiqarish).

Bu operatsiyalar navbatni to'g'ri boshqarishga va elementlarni tartib bilan chiqarib olishga yordam beradi.

Bundan tashqari, set (to'plam) va map (lug'at) kabi konteynerlar ham mavjud bo'lib, ular o'ziga xos ma'lumot tuzilmalari sifatida ma'lum maqsadlarga xizmat qiladi. Set noyob elementlarni saqlashda ishlatiladi va u elementlarning takrorlanishini oldini olish imkonini beradi. Set ichidagi ma'lumotlar tartiblangan bo'lmaydi, lekin undagi elementlarga tezkor kirish imkonini beradi. Map esa kalit-qiyamat juftligini saqlash uchun mo'ljallangan konteynerdir. Bu kalitlar orqali ma'lumotlarni tez va oson topish imkonini beradi, bu esa katta hajmdagi ma'lumotlar bilan ishlashda juda foydali.

Umuman olganda, konteynerlar va ular bilan ishlaydigan algoritmlar dasturlashning ajralmas qismi bo'lib, ular yordamida ma'lumotlarni samarali boshqarish va ularga ishlov berish imkoniyati yaratiladi. Ushbu maqolada konteynerlarning turlari va ularga mos keladigan asosiy algoritmlar haqida batafsil to'xtalib o'tamiz. Shuningdek, C++ dasturlash tilida konteynerlar bilan ishlash jarayoniga amaliy misollar orqali yondashamiz. Maqsad — konteynerlarning dasturlashdagi ahamiyatini va ularning samaradorligini tushuntirish.

Konteyner bilan ishlashga C++ dasturlash tilida misol:

```
#include <iostream>
#include <vector>
#include <algorithm> // sort funksiyasi uchun kerak
using namespace std;

int main() {
    // Vektorni yaratamiz va uni dastlabki qiymatlar bilan to'ldiramiz
    vector<int> numbers = {5, 2, 9, 1, 5, 6};

    // Dastlabki ro'yxatni chiqaramiz
    cout << "Dastlabki ro'yxat: ";
    for (int num : numbers) {
        cout << num << " ";
    }
    cout << endl;

    // Vektorni tartiblaymiz (sort funksiyasidan foydalanamiz)
    sort(numbers.begin(), numbers.end());
```

```
// Tartiblangan ro'yxatni chiqaramiz
cout << "Tartiblangan ro'yxat: ";
for (int num : numbers) {
    cout << num << " ";
}
cout << endl;

return 0;
}
```

Kodni tushuntirish

```
#include <vector> va #include <algorithm>
```

Biz dasturimizda vector konteynerini ishlatmoqdamiz, shuning uchun <vector> kutubxonasini dasturga kiritamiz. <algorithm> kutubxonasi esa sort funksiyasini ishlatish uchun kerak bo'ladi. sort — bu STL kutubxonasida mavjud bo'lgan tayyor algoritim bo'lib, u ma'lumotlarni tartiblashni avtomatik ravishda amalga oshiradi.

```
vector<int> numbers = {5, 2, 9, 1, 5, 6};
```

Bu yerda biz bir nechta butun sonlardan iborat vektor (yig'indi) yaratmoqdamiz. Vektor C++ STLda dinamik ravishda o'lchamini o'zgartira oladigan ma'lumotlar tuzilmasidir. Uning yordamida sonlarni saqlash va ularning ustida turli amallarni bajarish mumkin.

```
for (int num : numbers)
```

Bu sikl vektordagi har bir elementni ketma-ket ko'rib chiqadi va ularni ekranga chiqaradi. num har bir element qiymatini oladi va cout orqali ularni ko'rsatadi. Bu yerda dastlabki ro'yxat ekranga chiqariladi.

```
sort(numbers.begin(), numbers.end());
```

Bu qator vektor ichidagi barcha elementlarni kichikdan kattaga qarab tartiblaydi. sort funksiyasi ikkita parametr qabul qiladi: vektorning boshlanishi (numbers.begin()) va oxiri (numbers.end()). begin() vektorning birinchi elementiga ishora qiladi, end() esa so'nggi elementdan keyingi pozitsiyani bildiradi. Shu orqali butun vektor oralig'ida elementlar o'zaro taqqoslanadi va o'rinlari almashtiriladi.

Tartiblangan ro'yxatni chiqarish:

Kodning oxirida yana bir marta for sikli yordamida tartiblangan elementlar ekranga chiqariladi. Endi ro'yxat kichikdan kattaga qarab joylashgan bo'ladi.

Kod natijasi:

Dastlabki ro'yxat: 5 2 9 1 5 6

Tartiblangan ro'yxat: 1 2 5 5 6 9

Yuqoridagi kodda biz vector konteyneridan foydalandik va unga sort algoritmini qo'lladik. Bu kod oddiy bo'lishiga qaramasdan, C++ STL kutubxonasidagi kuchli vositalarni qanday ishlatishni ko'rsatadi. Vektor ma'lumotlarni saqlashda qulay bo'lsa, sort funksiyasi esa ularni samarali tartiblash imkonini beradi. sort funksiyasi katta hajmdagi ma'lumotlarni tez va samarali tartiblash uchun mo'ljallangan.

Xulosa. Konteynerlar va ular bilan ishlaydigan algoritmlar zamonaviy dasturlashda asosiy komponentlardan biri hisoblanadi. Ushbu maqolada biz konteynerlar va ular bilan ishlashning muhim tomonlarini, C++ dasturlash tilida STL kutubxonasi orqali konteynerlar bilan qanday ishlashni ko'rib chiqdik. Konteynerlar yordamida dasturlash jarayonida ma'lumotlarni saqlash va ularga ishlov berish sezilarli darajada soddalashadi va samaradorligi oshadi. Ma'lumotlarning har xil tuzilmalari bilan ishlashda mos keladigan konteynerlar tanlanadi, bu esa ma'lumotlarga tezkor va samarali kirishni ta'minlaydi.

Maqolada vector, stack, queue, set va map kabi asosiy konteynerlar bilan ishlash haqida ma'lumot berildi. Har bir konteynerning o'ziga xos xususiyatlari bor, ular muayyan ma'lumot turlarini boshqarish va amallarni bajarishda samaradorlikni ta'minlaydi. Vector — eng keng qo'llaniladigan konteynerlardan biri bo'lib, u dinamik ma'lumotlar yig'indisi sifatida har xil o'lchamdagi ma'lumotlarni saqlash imkonini beradi. Stack va queue ma'lumotlarga ketma-ket ishlov berish uchun ishlatiladi: stack LIFO (Last In, First Out) tamoyiliga asoslanib, so'nggi kiritilgan ma'lumotni birinchi bo'lib chiqaradi, queue esa FIFO (First In, First Out) tamoyiliga muvofiq ishlaydi va birinchi kiritilgan ma'lumot birinchi chiqariladi.

Shuningdek, set va map konteynerlari noyob elementlar va kalit-qiymat juftligini saqlashda samarali vositalar hisoblanadi. Set konteynerida takroriy elementlar bo'lmaydi, bu esa o'ziga xos qiymatlarni boshqarishni osonlashtiradi. Map esa ma'lumotlar juftligi sifatida kalitlar va qiymatlarni bog'lashda ishlatiladi, bu ma'lumotlarga tezkor kirish imkonini beradi.

Konteynerlar bilan ishlovchi algoritmlar, ayniqsa, tartiblash va qidirish amallarini samarali bajarishga mo'ljallangan. Sort kabi tayyor funksiyalar ma'lumotlarni tez va to'g'ri tartiblash imkonini beradi. Maqolada C++ tilida vector konteyneridan foydalanib, tartiblash algoritmi yordamida qanday qilib ma'lumotlarni oson tartiblash mumkinligi ko'rsatildi. Sort funksiyasi orqali vektor ichidagi elementlarni kichikdan kattaga qarab tartiblash jarayoni tushuntirildi. Ushbu funksiyadan foydalanish nafaqat kod yozish jarayonini tezlashtiradi, balki kodni osonroq tushunishga yordam beradi.

C++ dasturlash tilida konteynerlar va ularga mos algoritmlar yordamida murakkab ma'lumotlarni samarali boshqarish imkoniyati yaratiladi. STL kutubxonasi dasturchilarga nafaqat konteynerlarni, balki ular ustida ishlaydigan algoritmlarni ham taqdim etadi, bu esa kodni qisqartirish va optimallashtirish uchun qulay imkoniyatdir. Konteynerlar va ularning algoritmlari dasturiy ta'minotni ishlab chiqish jarayonida dasturchilarga ko'plab murakkab vazifalarni hal qilishda yordam beradi.

Umuman olganda, konteynerlar va ular bilan ishlaydigan algoritmlarning to'g'ri qo'llanilishi dasturiy ta'minotni yanada samarali va barqaror qiladi. Dasturchilar uchun konteynerlar bilan ishlashning asosiy tamoyillarini bilish va ularni turli amaliyotlarda qo'llay olish muhimdir. Ushbu maqolada keltirilgan nazariy va amaliy misollar konteynerlar bilan samarali ishlash uchun zarur bo'lgan bilimlarni taqdim etdi. Shu bilan birga, konteynerlar va algoritmlar yordamida dasturiy ta'minotni optimallashtirish hamda ma'lumotlar bilan ishlashda resurslardan to'g'ri foydalanish haqida kengroq tushuncha berdi.

FOYDALANILGAN ADABIYOTLAR:

1. Abdurakhmonova, N., Alisher, I., & Toirova, G. (2022, September). Applying Web Crawler Technologies for Compiling Parallel Corpora as one Stage of Natural Language Processing. In 2022 7th International Conference on Computer Science and Engineering (UBMK) (pp. 73-75). IEEE.
2. Abdurakhmonova, N., Alisher, I., & Sayfulleyeva, R. (2022, September). MorphUz: Morphological Analyzer for the Uzbek Language. In 2022 7th International Conference on Computer Science and Engineering (UBMK) (pp. 61-66). IEEE.
3. NAZIROVA, E., ABDURAKHMONOVA, N., & ALISHER, I. Exploring Linguistic Roots (stem) and Word Categories in Uzbek Language through Advanced Natural Language Processing Techniques for Text Analysis.
4. Ismailov, A. S., Akbarov, A., Qodirova, G. T. X. Q., & Yigitaliyeva, M. (2023). Dasturiy ta'minotni ishlab chiqish bosqichlari. Science and Education, 4(3), 187-191.