



SUN'IY INTELLEK VA MASHINALARGA OID MASALALARNI BUBBLE SORT SARALASH ALGORITIMI YORDAMIDA YECHISH

Farmonov Sherzodbek Raxmonjonovich

Farg'ona Davlat Universiteti amaliy matematika va informatika
kafedrası katta o'qituvchisi famonovsh@gmail.com

Mo'ydinova Asalxon Qodirjon qizi

Farg'ona Davlat Universiteti "Kompyuter ilmlari va dasturlash
texnologiyalari" yo'nalishi 23.11-guruh 2-bosqich talabasi
moydinovaasalxon181@gmail.com

<https://doi.org/10.5281/zenodo.14411632>

ARTICLE INFO

Qabul qilindi: 06- Dekabr 2024 yil
Ma'quillandi: 08- Dekabr 2024 yil
Nashr qilindi: 12- Dekabr 2024 yil

KEY WORDS

Bubble Sort, almashtirish, vaqt murakkabligi, samaradorlik, kichik ma'lumotlar to'plami, algoritim tamoyillari, ma'lumotlar ro'yxati.

ABSTRACT

Ushbu maqola C# dasturlash tilida Bubble Sort saralash algoritmini ko'rib chiqadi. Maqolada algoritimning asosiy tamoyillari, uning C# tilidagi amalga oshirilishi va ushbu saralash usulining afzalliklari va kamchiliklari batafsil tahlil qilinadi. Algoritimning ma'lumot hajmiga va amalga oshirish murakkabligiga qarab bajarilish vaqti haqida alohida e'tibor qaratiladi. Bunda biz kod misollari va samaradorlik testlari bilan tanishish imkoniyatiga ega bo'lib, algoritim qanday ishlashini va qaysi hollarda qo'llanilishi kerakligini yaxshiroq tushunishimiz mumkin.

Saralash algoritmlari ma'lumotlar to'plamini tartibga solish uchun ishlatiladi. Ularning ichida Bubble Sort algoritmi — ma'lumotlarni tartibga solishning eng oddiy va tushunarli usullaridan biri (pufakcha saralash) algoritmidir. Ushbu maqolada biz Bubble Sort algoritmini C# dasturlash tilida qanday amalga oshirishni ko'rib chiqamiz. Algoritimning ishlash prinsipi, kod namunasi va uning afzalliklari bilan kamchiliklarini muhokama qilamiz.

Bubble Sort Algoritmining Ishlash Prinsipi

Bubble Sort algoritmi oddiy bir tamoyilga asoslanadi: u ma'lumotlar ro'yxatini bir necha marta o'zaro solishtirib chiqadi va har safar eng katta (yoki eng kichik) elementni ro'yxatning oxiriga "pufak" kabi ko'taradi. Algoritim quyidagi qadamlarni bajaradi:

1. Ro'yxatning birinchi elementidan boshlab, qo'shni elementlar o'rtasida solishtirish amalga oshiriladi.
2. Agar birinchi element ikkinchi elementdan katta bo'lsa, u holda ularni almashtirish kerak.
3. Bu jarayon ro'yxatning oxirigacha davom etadi.
4. Har bir o'tishdan so'ng, eng katta element ro'yxatning oxiriga joylashtiriladi.
5. Ushbu jarayon, ro'yxat to'liq saralangan bo'lmaguncha takrorlanadi.

Algoritimning Pseudokod

Bubble Sort algoritmining pseudokodi quyidagicha ko'rinadi:

```
procedure bubbleSort(A: array of integers)
```

```
  n := length(A)
```

```
  for i from 0 to n-1 do
```

```
for j from 0 to n-i-2 do
    if A[j] > A[j+1] then
        swap A[j] and A[j+1]
```

Bu kodda A — saralash uchun berilgan massiv, n — massivning uzunligi. Algoritm har bir elementni solishtirib chiqadi va kerak bo'lsa, ularni almashtiradi.

Bubble Sort algoritmi, ro'yxatdagi elementlarni bir-biri bilan solishtirib, ularni kerakli tartibda joylashtirishga asoslangan. Har bir o'tish davomida eng katta (yoki eng kichik) element ro'yxatning oxiriga "pufak" kabi ko'tariladi. Bu jarayon ro'yxat to'liq saralangan bo'lmaguncha davom etadi.

Bubble Sort algoritmini tushunish uchun oddiy bir misol keltiraylik. Keling, 5ta sonni o'z ichiga olgan ro'yxatni saralash jarayonini ko'rib chiqamiz:

Ushbu masalalarda Bubble Sort saralash algoritmi oddiy va tushunarli ishlashi sababli foydalaniladi. Quyida Bubble Sort saralash algoritmiga oid masala keltirilgan:

Berilgan ro'yxat: [64, 34, 25, 12, 22]

O'tishlar:

Bubble Sort algoritmi har bir o'tishda ro'yxatning qo'shni elementlarini solishtiradi va kerak bo'lsa, ularni almashtiradi. Quyida har bir o'tish jarayonini ko'rib chiqamiz:

Dastlabki ro'yxat: [64, 34, 25, 12, 22]

• 1-o'tish:

- 64 va 34 ni solishtiramiz: $64 > 34$, almashtiramiz \rightarrow [34, 64, 25, 12, 22]
- 64 va 25 ni solishtiramiz: $64 > 25$, almashtiramiz \rightarrow [34, 25, 64, 12, 22]
- 64 va 12 ni solishtiramiz: $64 > 12$, almashtiramiz \rightarrow [34, 25, 12, 64, 22]
- 64 va 22 ni solishtiramiz: $64 > 22$, almashtiramiz \rightarrow [34, 25, 12, 22, 64]

Ro'yxat oxirida eng katta element (64) joylashdi.

• 2-o'tish:

- 34 va 25 ni solishtiramiz: $34 > 25$, almashtiramiz \rightarrow [25, 34, 12, 22, 64]
- 34 va 12 ni solishtiramiz: $34 > 12$, almashtiramiz \rightarrow [25, 12, 34, 22, 64]
- 34 va 22 ni solishtiramiz: $34 > 22$, almashtiramiz \rightarrow [25, 12, 22, 34, 64]

Ro'yxat oxirida eng katta element (34) joylashdi.

• 3-o'tish:

- 25 va 12 ni solishtiramiz: $25 > 12$, almashtiramiz \rightarrow [12, 25, 22, 34, 64]
- 25 va 22 ni solishtiramiz: $25 > 22$, almashtiramiz \rightarrow [12, 22, 25, 34, 64]

Ro'yxat oxirida eng katta element (25) joylashdi.

• 4-o'tish:

- 12 va 22 ni solishtiramiz: $12 < 22$, almashtirmaymiz.

Yechimni topish:

```
using System;
class Program
{
    static void Main(string[] args)
    {
        int[] numbers = { 64, 34, 25, 12, 22 };
        Console.WriteLine("Dastlabki massiv:");
        PrintArray(numbers);
    }
}
```

```

BubbleSort(numbers);
Console.WriteLine("Saralangan massiv:");
PrintArray(numbers);
}
static void BubbleSort(int[] arr)
{
    int n = arr.Length;
    for (int i = 0; i < n - 1; i++)
    {
        // Oxirgi i ta elementlar allaqachon saralangan
        for (int j = 0; j < n - i - 1; j++)
        {
            if (arr[j] > arr[j + 1])
            {
                // Elementlarni almashtirish
                Swap(ref arr[j], ref arr[j + 1]);
            }
        }
    }
}

static void Swap(ref int a, ref int b)
{
    int temp = a;
    a = b;
    b = temp;
}

static void PrintArray(int[] arr)
{
    foreach (int num in arr)
    {
        Console.Write(num + " ");
    }
    Console.WriteLine();
}
}

```



Natija:

Endi ro'yxat to'liq saralangan va natijaviy ro'yxat quyidagicha bo'ladi:
Saralangan ro'yxat: [12, 22, 25, 34, 64]

Kodni Tushuntirish

1. Massivni e'lon qilish: `int[] numbers` massivini e'lon qilib, saralash uchun berilgan sonlar to'plamini kiritamiz.

2. PrintArray metodi: Ushbu metod massivni konsolga chiqaradi.

3. BubbleSort metodi: Bu metod Bubble Sort algoritmini amalga oshiradi:

- Dastlab massiv uzunligini n o'zgaruvchisiga saqlaymiz.
- Ikkita for sikli yordamida massiv elementlarini solishtirib boramiz.
- Agar j -chi element i -chi elementdan katta bo'lsa, Swap metodini chaqirib, ularni almashtiramiz.

4. Swap metodi: Bu metod ikki elementni almashtirish uchun ishlatiladi. `ref` kalit so'zlari yordamida o'zgaruvchilarni to'g'ridan-to'g'ri almashtiramiz.

Afzalliklari

• **Soddaligi:** Bubble Sort algoritmi juda oddiy va tushunarli. Uni o'rganish va amalga oshirish oson.

• **In-place saralash:** Bu algoritm qo'shimcha xotira talab qilmaydi, ya'ni saralash jarayonida yangi massiv yaratmaydi.

• **Stabiliti:** Bubble Sort stabil saralash algoritmi hisoblanadi, ya'ni bir xil kalitlarga ega elementlar o'z joylarini saqlaydi.

Kamchiliklari

• **Tezlik:** Bubble Sort algoritmi n^2 vaqt murakkabligiga ega, bu esa katta ma'lumotlar to'plami uchun juda sekin bo'lishi mumkin. Buning natijasida u amaliyotda ko'pincha samarali saralash algoritmlari bilan almashtiriladi.

• **Ishlash samaradorligi:** Katta ma'lumotlar bilan ishlaganda, Bubble Sort algoritmi ko'pincha boshqa murakkab saralash usullaridan ancha past samaradorlikka ega.

Amaliy Qo'llanilishi

Bubble Sort algoritmi ko'pincha ta'lim jarayonida o'rganiladi, chunki u boshqa murakkab saralash algoritmlarini tushunish uchun asos bo'lib xizmat qiladi. Shuningdek, kichik ma'lumotlar to'plamlari bilan ishlaganda yoki tezkor prototiplar yaratishda foydalanilishi mumkin.

Xulosa

C# dasturlash tilida Bubble Sort algoritmi sodda va tushunarli bo'lib, dasturchilar uchun yaxshi o'rganish vositasi hisoblanadi. Garchi uning samaradorligi yuqori bo'lmasa-da, bu algoritm dasturlash asoslarini o'rganishda muhim ahamiyatga ega. Katta ma'lumotlar to'plami bilan ishlaganda boshqa samarali saralash usullarini (masalan, Quick Sort yoki Merge Sort) qo'llash tavsiya etiladi. Bubble Sortning o'ziga xosligi va soddaligi uni dasturlash ta'limida ajralmas vosita sifatida saqlab qoladi.

Foydalanilgan adabiyotlar ro'yhati:

1. "Introduction to Algorithms" - Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein
2. "Algorithms" - Robert Sedgewick, Kevin Wayne
3. "Data Structures and Algorithms Made Easy" - Narasimha Karumanchi
4. GeeksforGeeks (<https://www.geeksforgeeks.org>)
5. Wikipedia (https://en.wikipedia.org/wiki/Bubble_sort)