

Практична робота № 6

Тема: Реалізація інтерполяційного пошуку

Мета роботи: закріплення навичок роботи з масивами, засвоєння швидких алгоритмів пошуку.

Теоретичні відомості

Бінарний пошук кожен раз шукає цільовий елемент масиву в середині розділу, інтерполяційний намагається прискорити процес - він намагається вгадати розташування цільового елемента в масиві по його значенню. Припустимо, в масиві є 1000 елементів зі значеннями від 1 до 100. Якщо наша мета - знайти число 30, то його потрібно шукати в районі першої третини масиву, деє поруч з індексом 300.

Загальний розподіл чисел не завжди дозволяє отримати результат з прицільною точністю, але він може виявитися досить близьким.

На рис. 1 зображений інтерполяційний пошук числа 77.

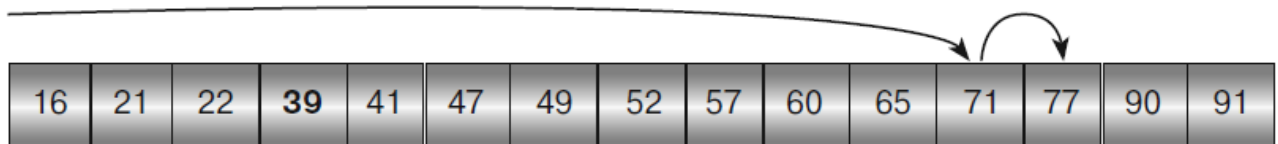


Рис. 1 – Інтерполяційний пошук

Наступний псевдокод є високорівневий алгоритм даного методу.

```
Integer: InterpolationSearch(Data values[], Data target)
Integer: min = 0
Integer: max = values.Length - 1
While (min <= max)
    // Знаходимо цільовий елемент.
    Integer: mid = min + (max - min) *
        (target - values[min]) / (values[max] - values[min])
    If (values[mid] == target) Then Return mid
    If ((mid < min) || (mid > max)) Return 0;
    // See if we need to search the left or right half.
```

```

    If (values[mid] < target) min = mid + 1;
    Else if (values[mid] > target) max = mid - 1;
    Else Return;
End While
Return 0
End InterpolationSearch

```

Найскладніша частина даного коду - оператор для розрахунку mid. Щоб би знайти його значення, до поточного значення min додається відстань від min до max, яка масштабується по очікуваній частці відстані від values [min] до values [max], де повинен знаходитися target. Якщо дані розподілені дуже нерівномірно і ви шукаєте найгірше значення, даний алгоритм має продуктивність $O(N)$. Якщо розподіл рівномірний, очікувана продуктивність складе $O(\log(\log N))$.

Завдання

1. Запустити програму InterpolationSearch (рис. 2). Згенерувати послідовність $30 + N$ чисел від 1 до 1000, де N – номер студента за навчальним журналом.

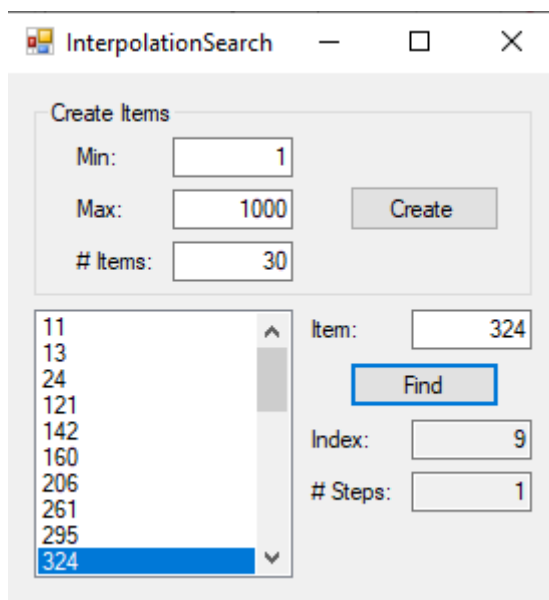


Рис. 2 – Робота програми InterpolationSearch

2. Написати програму, в якій:

- сформувати одновимірний масив цілих, що з генеровані у п.1;
- виконати друк цього масиву на екран.

- здійснити пошук вказаного (з клавіатури) елемента у масиві, використовуючи інтерполяційний пошук;
- вивести на екран індекс елемента у масиві (Index) та кількість кроків пошуку (Steps) та порівняти з даними програми InterpolationSearch (на рис. 2 показано, що здійснюється пошук числа 324. Індекс числа 9, кількість кроків пошуку – 1). При відсутності числа у масиві вивести «None»

3. Оформити звіт. Навести результати роботи програми InterpolationSearch та програми, що створена у п.2 (див. рис. 3).

```
clang version 7.0.0-3~ubuntu0.18.04.1 (tags/RELEASE_700/final)
➤ clang++-7 -pthread -o main main.cpp
➤ ./main
04 06 09 10 13 13 16 18 20 24 27 29 36 36 38 42 44 45 45 52 60
   63 66 72 73 77 87 90 94 99
Загадайте число від 1 до 100:
88

Number: 88

none

04 06 09 10 13 13 16 18 20 24 27 29 36 36 38 42 44 45 45 52 60
   63 66 72 73 77 87 90 94 99
Загадайте число від 1 до 100:
45

Number: 45

key: 17

Steps: 4
```

Рис. 3 – Приклад роботи програми інтерполяційного пошуку

Контрольні питання

1. Які алгоритми пошуку є? Назвіть їх принцип роботи та орієнтовний час роботи.
2. Чим відрізняється інтерполяційний пошук від бінарного?
3. Який час пошуку елемента лінійним, бінарним, інтерполяційним методом у послідовності з 1000000 чисел?
4. Чи потрібно виконувати сортування даних у масиві перед початком пошуку?