PDP
IT-ACADEMY

Qidirish algoritmlari

PDP
IT-ACADEMY

Reja:

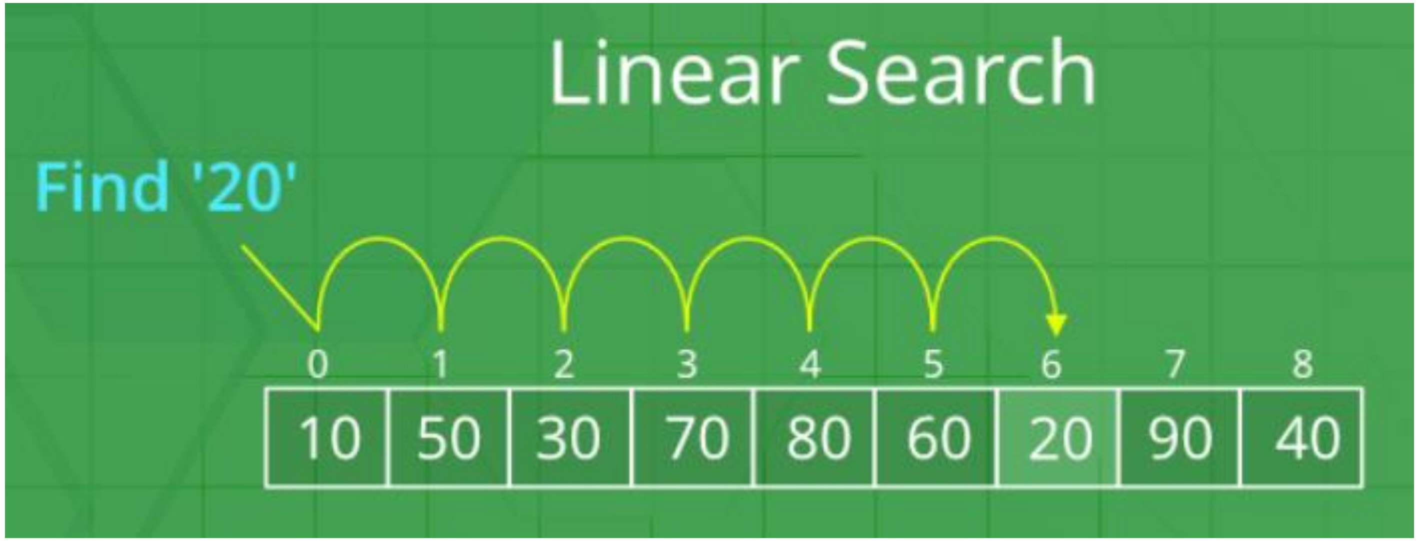
- LINEAR SEARCH
- BINARY SEARCH
- JUMP SEARCH

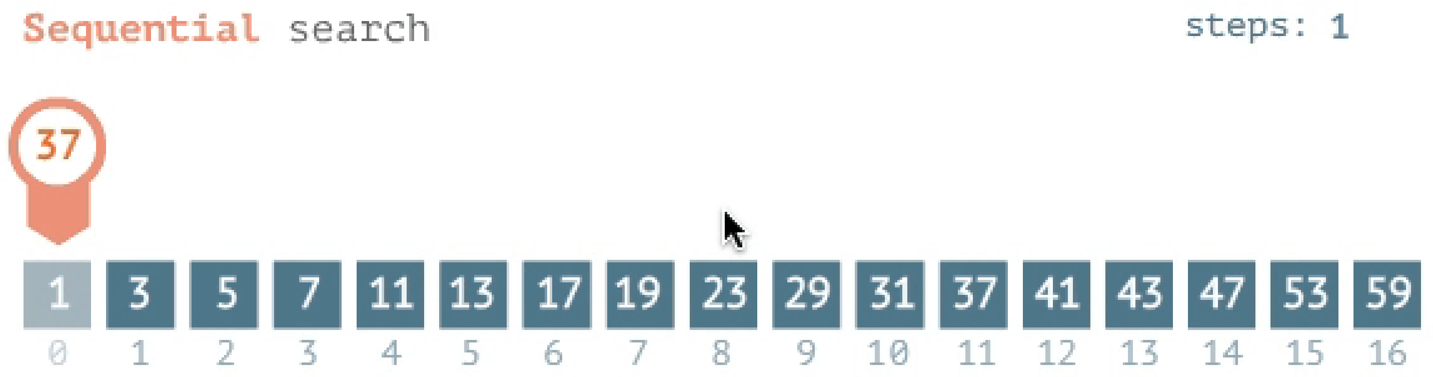
PDP
IT-ACADEMY

PDP
IT-ACADEMY

LINEAR SEARCH

PDP
IT-ACADEMY




Sequential search
steps: 1
37
1 3 5 7 11 13 17 19 23 29 31 37 41 43 47 53 59
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16

```
// Linearly search x in arr[].
// If x is present then return its
// location,  otherwise return -1

int linearSearch(int arr[], int n, int x)
{
    for (int i = 0; i < n; i++)
        if (arr[i] == x)
            return i;

    return -1;
}
```

PDP
IT-ACADEMY

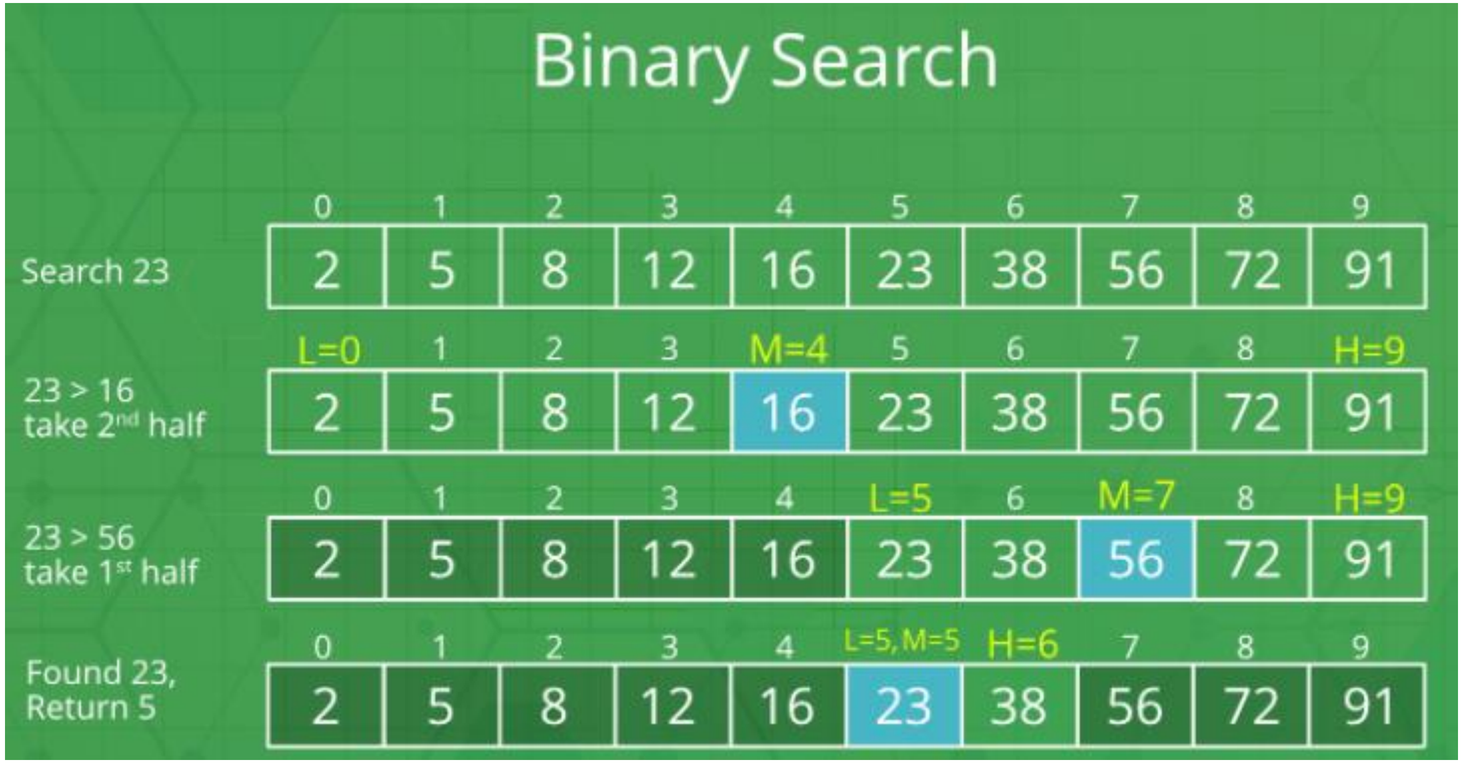
```cpp
int main()
{

    int arr[] = { 3, 4, 1, 7, 5 };
    int n = sizeof(arr) / sizeof(arr[0]);
    int x = 4;

    int index = linearSearch(arr, n, x);

    if (index == -1)
        cout << "Element is not present in the array"<<endl;
    else
        cout << "Element found at position " << index<<endl;

    return 0;
}
```

```
Element found at position 1
```

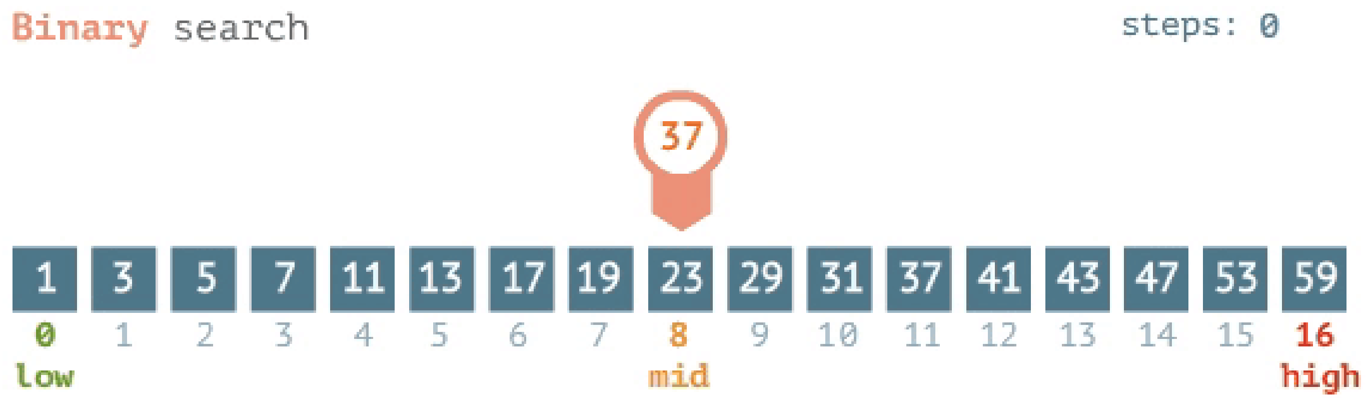
PDP
IT-ACADEMY

BINARY SEARCH

PDP IT-ACADEMY



PDP
IT-ACADEMY



PDP IT-ACADEMY

```
// A iterative binary search function. It returns
// location of x in given array arr[left..right] if present,
// otherwise -1
int binarySearch(int arr[], int left, int right, int x)
{
    while (left <= right) {
        int middle = left + (right - left) / 2;

        // Check if x is present at mid
        if (arr[middle] == x)
            return middle;

        // If x greater, ignore left half
        if (arr[middle] < x)
            left = middle + 1;

        // If x is smaller, ignore right half
        else
            right = middle - 1;
    }

    // if we reach here, then element was
    // not present
    return -1;
}
```

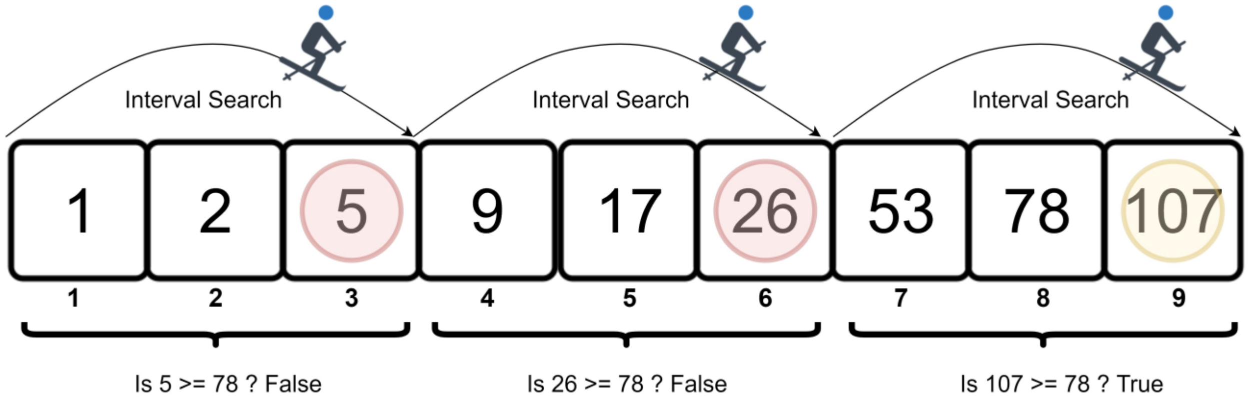
PDP
IT-ACADEMY

```cpp
int main()
{

    int arr[] = { 2, 3, 4, 10, 40 };
    int x = 10;

    int n = sizeof(arr) / sizeof(arr[0]);

    int result = binarySearch(arr, 0, n - 1, x);

    (result == -1) ? cout << "Element is not present in array"
                   : cout << "Element is present at index " << result;


    cout<<endl;
    return 0;
}
```

```
Element is present at index 3
```

PDP
IT-ACADEMY

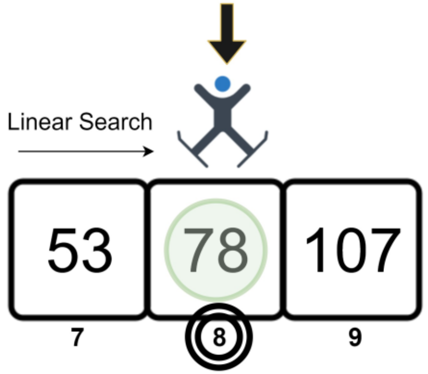
JUMP SEARCH



n = len(A) = 9

m = √n = 3

item to be found = 78

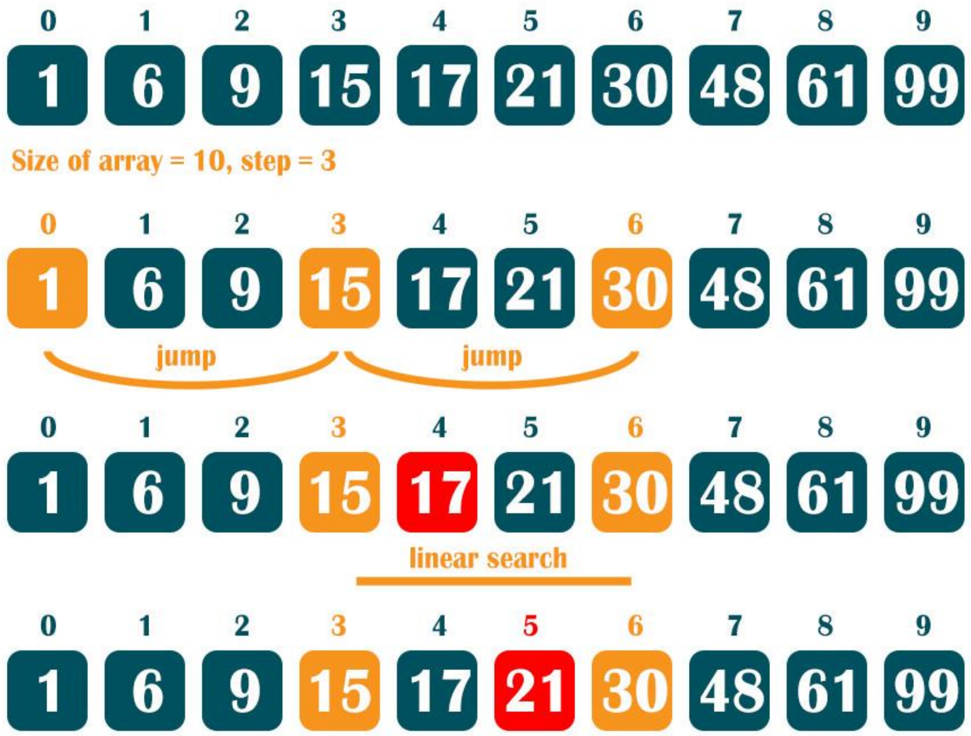


Index of 78 is 8

PDP IT-ACADEMY

Search for 21

Jump Search



Element found at index 5

PDP
IT-ACADEMY

```
int jumpSearch(int arr[], int n, int x)
{
    // Finding block size to be jumped
    int step = sqrt(n);

    // Finding the block where element is
    // present (if it is present)
    int prev = 0;
    while (arr[min(step, n)-1] < x)
    {
        prev = step;
        step += sqrt(n);
        if (prev >= n)
            return -1;
    }
```

PDP IT-ACADEMY

```
    // Doing a linear search for x in block
    // beginning with prev.
    while (arr[prev] < x)
    {
        prev++;

        // If we reached next block or end of
        // array, element is not present.
        if (prev == min(step, n))
            return -1;
    }
    // If element is found
    if (arr[prev] == x)
        return prev;

    return -1;
}
```

PDP IT-ACADEMY

```
int main()
{

    int arr[] = { 0, 1, 1, 2, 3, 5, 8, 13, 21,
                 34, 55, 89, 144, 233, 377, 610 };

    int x = 55;
    int n = sizeof(arr) / sizeof(arr[0]);

    // Find the index of 'x' using Jump Search
    int index = jumpSearch(arr, n, x);

    // Print the index where 'x' is located
    cout << "\nNumber " << x << " is at index " << index;

    cout<<endl;
    return 0;
}
```

```
Number 55 is at index 10
```

Amaliy mashq

PDP
IT-ACADEMY

“Son topish o‘yini” dasturini tuzish

PDP
IT-ACADEMY

Qiziqarli loyihalar

PDP
IT-ACADEMY

“Omad Lotto o’yini” dasturini tuzish

PDP
IT-ACADEMY

Restoran uchun “Buyurtmalar loyihasi” ning dasturini tuzish. Bu dastur menyusida kamida quyidagi amallar bo’lishi kerak:

- Restoran menyusini ko’rish;
- Taom buyurtma qilish;
- Ichimlik buyurtma qilish;
- Buyurtma uchun hisobni aniqlash (chek chiqarish)

PDP
IT-ACADEMY

5 qavatli, har bir qavatida 20 avtomobil saqlash joyi mavjud bo'lgan avtoturargoh bor. Quyidagi amallarni funksiyalar orqali bajaring:

1) Avtoturargohdagi band joylar sonini aniqlash
2) Avtoturargohdagi bo'sh joylar sonini aniqlash
3) Har bir qavatdagi bo'sh joylarning o'rnini ekranga chiqarish
4) Avtoturargohga avtomobil joylashtirish
5) Avtoturargohdan avtomobilni chiqarish

PDP
IT-ACADEMY

E`tiboringiz uchun rahmat!