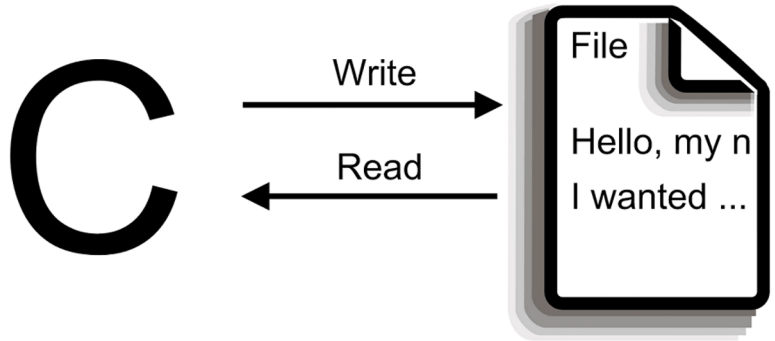
FILE HANDLING IN C

Searching and sorting

FILE

- Storage of data in variables and arrays is temporary such data is lost when a program terminates.
- Files are used for permanent retention of data.
- Computers store files on secondary storage devices, such as hard drives, CDs, DVDs and flash drives.



Text files vs Binary files

- Text file is a term used for a file that is essentially a sequence of character codes.

- Binary file is a term used for a file in which most byte are not intended to be interpreted as character codes.
- Here are a few common binary file formats:
- PDF, for documents
- JPEG, GIF, and PNG, for images
- MP3, for audio tracks

Steps to create and open a file

- Create file by a pointer variable using the FILE structure:
- FILE *p;
- Open the file, associating the file name.
- Read or write the data.
- Close the file.

File operations

Function	Operation
fopen()	Creates a new file / opens an existing file
fclose()	Closes a file which has been opened for use
getc()	Reads a character from the file
putc()	Writes a character to the file
fprintf()	Write data values to a file
fscanf()	Reads a set of data values from a file
getw()	Reads an integer from the file
putw()	Writes an integer to a file
fseek()	Sets the position to the desired point in the file
ftell()	Gives the current position in the file
rewind()	Sets the position to the beginning of the file

Open the file using : fopen()

FILE *fopen(const char *filename, const char *mode);

Using Files in C

Declaration of File Pointer

```
FILE *fp;
FILE *fp1;
```

Opening a File

```
fp=fopen("one.txt","r");
```

Checking the result of fopen()

```
if(fp == NULL)
{
    printf("Can't open");
    exit(1);
}
```

FILE OPENING MODES

"r" Opens a text file in reading mode

"w" Opens or create a text file in writing mode

"a" Opens a text file in append mode

"r+" Opens a text file in both reading and writing mode

"w+" Opens a text file in both reading and writing mode

"a+" Opens a binary file in reading mode

"rb" Opens a binary file in reading mode

"wb" Opens or create a binary file in writing mode

FUNCTIONS TO READ AND WRITE DATA TO FILE

- Function fprintf
  - Like printf
  - Takes first argument as file pointer

- Function fscanf
  - Like scanf
  - Takes first argument as file pointer

Random Access of Files

Function	Purpose
fseek ( )	To move the file pointer to a specific location.
ftell ( )	Returns the current value of the file pointer.
rewind ( )	To reset the file pointer and put it at the beginning of the file.

- fseek()
  - It sets the file pointer associated with file handle to a new position
  - This function is used to move the file pointer to different positions
  - fseek(fptr,offset,mode);
  - Where
    - fptr is a file pointer
    - offset is an integer that specifies the number of bytes by which file pointer is moved
    - mode specifies from which position the offset is measured
      - Its value is 0,1,2

Functions used in random access

1. *ftell()*: This function takes a file pointer as argument and returns a number of type long, that indicates the current position of the file pointer within the file. This function is useful in saving the current position of a file, which can be used later in the program.

Syntax

n = ftell(fp);

Here, n would give the relative offset (in bytes) of the current position. This means that n bytes have already been read (or written).

62

SEARCHING

- Searching and sorting are the two essential aspects of computer science.
- In Searching we search for a particular key or a file
- Two types of searching
  - Linear search
  - Binary search

LINEAR SEARCH

- Linear search is also known as sequential search, where we search an key against an array in a linear.
- Algorithm for searching
- linear_search(a[0…..n],n,key)
- {

for(i=0 to n)

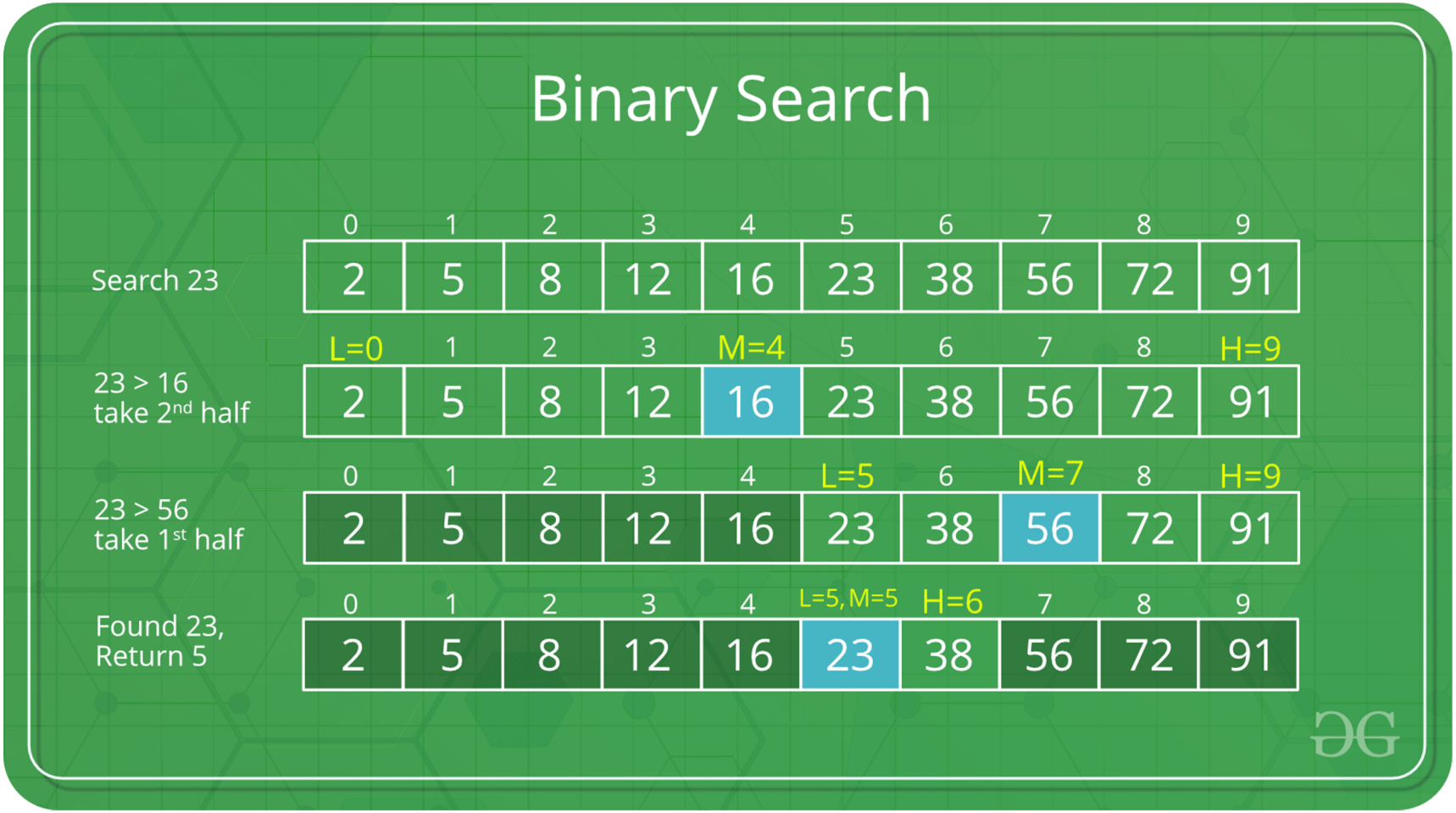
if(a[i] = key)

search is successful

- }

BINARY SEARCH

- One of the most efficient search algorithm.
- Here we divide the array in to two half and search for a particular key
- Array should be always sorted
- Each time an array with n elements is divided into
- n/2,n/4,n/8……..n/2^k times



Source geeksforgeeks

Binary search algorithm

```
binary_search(A,n,key)
{
        b=0,e=n-1
        while(b<e)
        {
                Mid = (b+e)/2
                if(a[mid]==key)
                        Search found
                elseif(key>a[mid])
                        b=mid+1
                else
                        e=mid-1
        }
}
```

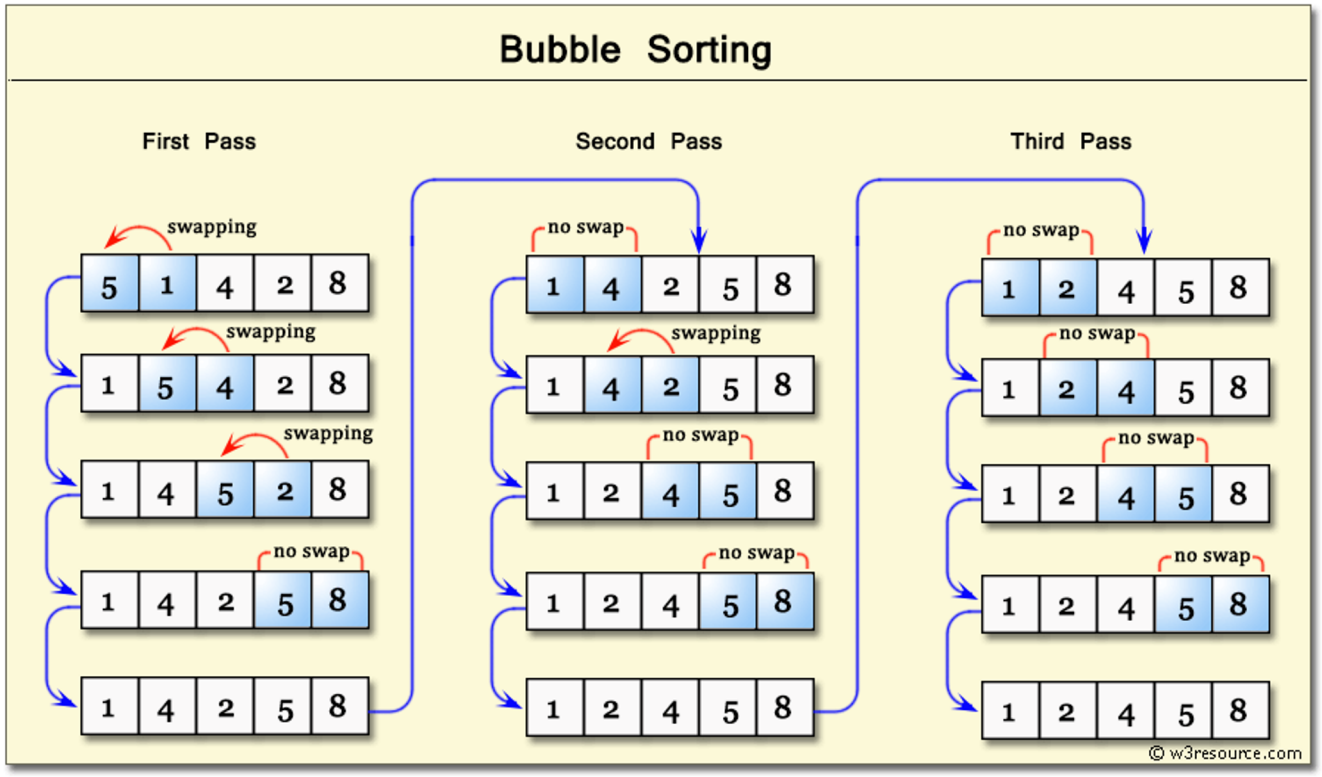
SORTING

- Arranging elements in order is called sorting
- Sorting helps to search an element more effectively
- There are many techniques in sorting

- **Bubble sort**
- **Selection sort**
- Insertion sort
- Merge sort
- Quick sort
- Radix sort

BUBBLE SORT

This is a brute force techniques , where we compare each element with next element and arrange in order

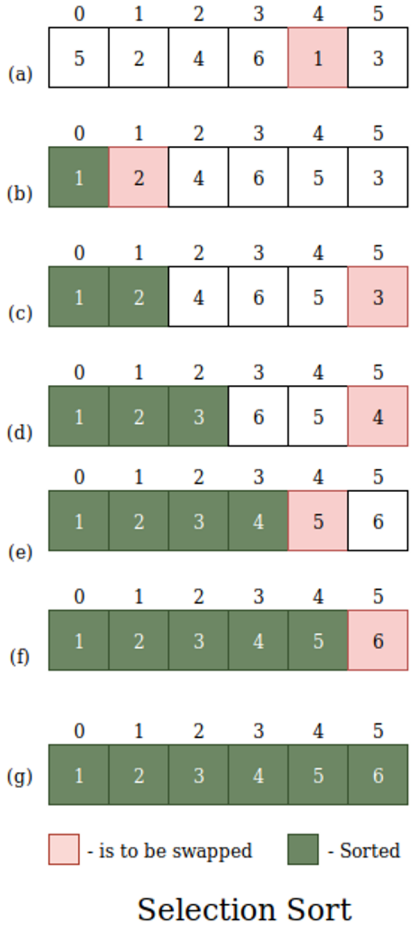


BUBBLE SORT Algorithm

```
bubble_sort(a[0...n],n)//n is the size of the array
{
        for(i=0 to n)
        {
                for(j=0 to n-i-1)
                {
                        if(a[j] > a[j+1])
                                swap(a[j] with a[j+1])
                }
        }
}
```

SELECTION SORT

As the name says select the element and swap with starting element.



Selection Sort

SELECTION SORT ALGORITHM

```
selection_sort(a[0....n], n)
{
        for(i=0 to n)
        {
                for(j=0 to n)
                {
                        if(min(a[j]))
                }
                swap(min with a[i])
        }
}
```