



BINARY SEARCH

1-USING ARRAYS

2-USING LINKLIST

3-USING DOUBLE LINKLIST

By DJVprogrammers



BINARY SEARCH USING ARRAYS

Code:

```
#include<iostream>

using namespace std;

class searching
{
    int arr[100];

    int size,num,loc,mid;

    public:

        searching()
        {
            cout<<"Enter Size of Array"<<endl;

            cin>>size;

            arr[size];

            mid=(1+size)/2;

            loc=-1;

            input();

        }

        void input()

        {
```

```
        cout<<"\nEnter numbers in array\n"<<endl;

        for(int i=0;i<size;i++)

            cin>>arr[i];

        search_no();

    }

    void search_no()

    {

        cout<<"\nEnter Number you want to search from array"<<endl;

        cin>>num;

        sorting();

    }

    void sorting()

    {

        int temp,i,j;

        for(i=0;i<size;i++)

        {

            for(j=i;j<size;j++)

            {

                if(arr[i]>=arr[j])

                {

                    temp=arr[i];
```

```
        arr[i]=arr[j];

        arr[j]=temp;

    }

}

}

cout<<"\nSORTED DATA\n";

for(i=0;i<size;i++)

{

    cout<<arr[i]<<" ";

}

cout<<endl;

binary_search();

}

void binary_search()

{

    if(arr[mid]==num)

    {

        loc=mid+1;

    }

    else if(arr[mid]>num)

    {
```

```
        for(int i=mid-1;i>=0;i--)
        {
            if(num==arr[i])
            {
                loc=i+1;
                break;
            }
        }
    }
    else
    {
        for(int i=mid;i<size;i++)
        {
            if(num==arr[i])
            {
                loc=i+1;
                break;
            }
        }
    }
}
```

```
    }

    ~searching()

    {

        if(loc==-1)

            cout<<"\nNumber not found "<<endl;

        else

            cout<<"\nNumber found at "<<loc<<endl;

    }

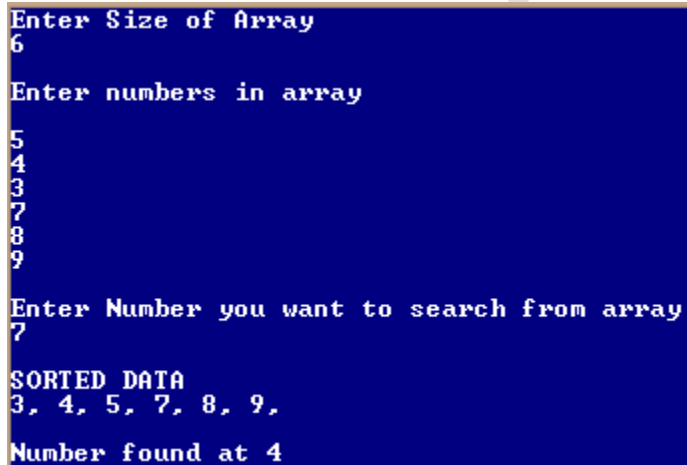
};

int main()

{

    searching s1;

}
```

A screenshot of a C++ program execution. The background is dark blue with white text. The text shows the program's output: 'Enter Size of Array' followed by '6', 'Enter numbers in array' followed by '5', '4', '3', '7', '8', '9', 'Enter Number you want to search from array' followed by '7', 'SORTED DATA' followed by '3, 4, 5, 7, 8, 9,', and 'Number found at 4'.

```
Enter Size of Array
6
Enter numbers in array
5
4
3
7
8
9
Enter Number you want to search from array
7
SORTED DATA
3, 4, 5, 7, 8, 9,
Number found at 4
```

BINARY SEARCH USING LINKLIST

Code

```
#include<iostream>
```

```
using namespace std;
```

```
struct node
```

```
{
```

```
    int data;
```

```
    node*link;
```

```
    node()
```

```
    {
```

```
        link=NULL;
```

```
    }
```

```
};
```

```
class linklist
```

```
{
```

```
    node*start,*cur,*temp;
```

```
    public:
```

```
    linklist()
```

```
    {
```

```
        start=NULL;
```

```
    }
```

```
creat_node(int n)
{
    if(start==NULL)
    {
        start=new node;
        start->link=NULL;
        start->data=n;
    }
    else
    {
        cur=start;
        while(cur->link!=NULL)
        {
            cur=cur->link;
        }
        temp=new node;
        temp->data=n;
        temp->link=NULL;
        cur->link=temp;
    }
}
```

```
//selection sorting

void sorting()

{

    cur=temp=start;

    linklist*add;

    int t;

    while(cur->link!=NULL)//outer loop

    {

        temp=cur;

        while(temp->link!=NULL)//inner loop

        {

            if(cur->data>=temp->data)

            {

                t=cur->data;

                cur->data=temp->data;

                temp->data=t;

            }

            temp=temp->link;

        }

        //used this if to compare data of last node
```

```
        if(cur->data>=temp->data)
        {
            t=cur->data;

            cur->data=temp->data;

            temp->data=t;
        }

        cur=cur->link;
    }

}

//print data of the linklist.

void print()
{
    cout<<"_____ "<<endl;

    cout<<"_____SORTED DATA IS_____ "<<endl;

    cur=start;

    int c;

    for(c=1;cur->link!=NULL;c++)
    {

        cout<<"Data "<<c<<" ---> "<<cur->data<<endl;
```

```
        cur=cur->link;

    }

    cout<<"Data "<<c<<" ---> "<<cur->data<<endl;

}

void searching(int s,int size)

{

    int mid,c=0;

    int check=0;

    cur=start;

    //finding middle value

    while(c!=(size)/2)

    {

        cur=cur->link;

        c++;

    }

    mid=cur->data;

    cout<<"\nMiddle value is "<<mid<<endl<<endl;

    //this condition is for the case when data found at middle

    if(mid==s)

    {
```

```
        check=1;

        cout<<"Data found at node no. "<<c+1<<endl;

    }

    // if Data to be search is greater then middle value of sorted data

    //then we will check the nodes from mid to the end

    else if(mid<s)

    {

        temp=cur;

        //loop to find the number

        for(c=c+1;temp->link!=NULL;c++)

        {

            if(temp->data==s)

            {

                check=1;

                cout<<"Data found at Node "<<c<<endl;

            }

            temp=temp->link;

        }

        //In linked list we need to apply seperate conditions for last node

        if(temp->data==s)

        {
```

```
        check=1;

        cout<<"Data found at Node "<<c<<endl;

    }

}

//if data to be searched is less then the middle value of sorted data

//then we will check the linklist from start to the mid..

else

{

    temp=start;

    //loop to check data... c is counter to find no. of the node where we find

data

    for(c=1;temp->link!=cur;c++)

    {

        if(temp->data==s)

        {

            check=1;

            cout<<"Data found at node "<<c<<endl;

        }

        temp=temp->link;

    }

    //to check the last value
```

<https://web.facebook.com/DJVprogrammers>

```
    cout<<"how many Nodes you want to create"<<endl;

    cin>>noofnodes;

    //creating nodes(function call and passing data

    for(i=1;i<=noofnodes;i++)

    {

        cout<<"Enter data:\t";

        cin>>n;

        l1.creat_node(n);

    }

    cout<<"\n\t=====Sorting
Data===== "<<endl<<endl;

    l1.sorting();

    l1.print();

    cout<<"\n\t=====SEARCHING
NUMBER===== \n\n";

    cout<<"Enter data you want to search"<<endl;

    int s;

    cin>>s;

    l1.searching(s,noofnodes);

}
```

```
=====
                        Binary Search using Linked list
=====

how many Nodes you want to create
6
Enter data:      5
Enter data:      3
Enter data:      2
Enter data:      2
Enter data:      9
Enter data:      2

=====Sorting Data=====

                        SORTED DATA IS_____
Data 1 ---> 2
Data 2 ---> 2
Data 3 ---> 2
Data 4 ---> 3
Data 5 ---> 5
Data 6 ---> 9

=====SEARCHING NUMBER=====

Enter data you want to search
2

Middle value is 3

Data found at node 1
Data found at node 2
Data found at node 3
```

```
=====
                        Binary Search using Linked list
=====

how many Nodes you want to create
5
Enter data:      4
Enter data:      2
Enter data:      1
Enter data:      7
Enter data:      8

=====Sorting Data=====

                        SORTED DATA IS_____
Data 1 ---> 1
Data 2 ---> 2
Data 3 ---> 4
Data 4 ---> 7
Data 5 ---> 8

=====SEARCHING NUMBER=====

Enter data you want to search
7

Middle value is 4

Data found at Node 4
```

BINARY SEARCH USING DOUBLE LINKED LIST

Code:

```
#include<iostream>
```

```
using namespace std;
```

```
struct node
```

```
{
```

```
    int data;
```

```
    node*link, *pre;
```

```
    node()
```

```
    {
```

```
        link=NULL;
```

```
    }
```

```
};
```

```
class linklist
```

```
{
```

```
    node*start,*cur,*temp;
```

```
    public:
```

```
    linklist()
```

```
    {
```

```
        start=NULL;
```

```
}

creat_node(int n)

{

    if(start==NULL)

    {

        start=new node;

        start->pre=NULL;

        start->link=NULL;

        start->data=n;

    }

    else

    {

        cur=start;

        while(cur->link!=NULL)

        {

            cur=cur->link;

        }

        temp=new node;

        temp->data=n;

        temp->pre=cur;

        temp->link=NULL;
```

```
        cur->link=temp;

    }

}

//selection sorting

void sorting()

{

    cur=temp=start;

    linklist*add;

    int t;

    while(cur->link!=NULL)//outer loop

    {

        temp=cur;

        while(temp->link!=NULL)//inner loop

        {

            if(cur->data>=temp->data)

            {

                t=cur->data;

                cur->data=temp->data;

                temp->data=t;

            }

        }

    }

}
```

```
        temp=temp->link;

    }

    //used this if to compare data of last node

    if(cur->data>=temp->data)

    {

        t=cur->data;

        cur->data=temp->data;

        temp->data=t;

    }

    cur=cur->link;

}

}

//print data of the linklist.

void print()

{

    cout<<"_____ "<<endl;

    cout<<"_____SORTED DATA IS _____ "<<endl;

    cur=start;

    int c;
```

```
for(c=1;cur->link!=NULL;c++)  
{  
  
    cout<<"Data "<<c<<" ---> "<<cur->data<<endl;  
  
    cur=cur->link;  
  
}  
  
cout<<"Data "<<c<<" ---> "<<cur->data<<endl;  
  
}  
  
void searching(int s,int size)  
{  
  
    int mid,c=0;  
  
    int check=0;  
  
    cur=start;  
  
    //finding middle value  
    while(c!=(size)/2)  
    {  
  
        cur=cur->link;  
  
        c++;  
  
    }  
  
    mid=cur->data;  
  
    cout<<"\nMiddle value is "<<mid<<endl<<endl;
```

```
//this condition is for the case when data found at middle

if(mid==s)

{

    check=1;

    cout<<"Data found at node no. "<<c+1<<endl;

}

// if Data to be search is greater then middle value of sorted data

//then we will check the nodes from mid to the end

else if(mid<s)

{

    temp=cur;

    //loop to find the number

    for(c=c+1;temp->link!=NULL;c++)

    {

        if(temp->data==s)

        {

            check=1;

            cout<<"Data found at Node "<<c<<endl;

        }

        temp=temp->link;

    }

}
```

//In linked list we need to apply separate conditions for last node

```
if(temp->data==s)
```

```
{
```

```
    check=1;
```

```
    cout<<"Data found at Node "<<c<<endl;
```

```
}
```

```
}
```

//if data to be searched is less than the middle value of sorted data

//then we will check the linklist from mid to start..

```
else
```

```
{
```

```
    temp=cur;;
```

//loop to check data... c is counter to find no. of the node where we find

data

```
    for(c=c+1;temp->pre!=NULL;c--)
```

```
    {
```

```
        if(temp->data==s)
```

```
        {
```

```
            check=1;
```

```
            cout<<"Data found at node "<<c<<endl;
```

```
        }
```

<https://web.facebook.com/DJVprogrammers>

```
        cout<<"\t=====
=\\n"<<endl;

        int noofnodes,i,n;

        //no. of nodes

        cout<<"how many Nodes you want to create"<<endl;

        cin>>noofnodes;

        //creating nodes(function call and passing data
        for(i=1;i<=noofnodes;i++)
        {

                cout<<"Enter data:\\t";

                cin>>n;

                l1.creat_node(n);

        }

        cout<<"\\n\\t=====Sorting
Data===== "<<endl<<endl;

        l1.sorting();

        l1.print();

        cout<<"\\n\\t=====SEARCHING
NUMBER=====\\n\\n";

        cout<<"Enter data you want to search"<<endl;

        int s;

        cin>>s;
```

```
l1.searching(s,noofnodes);  
  
}
```

```
=====
                        Binary Search using Linked list
=====

how many Nodes you want to create
6
Enter data:      5
Enter data:      4
Enter data:      3
Enter data:      2
Enter data:      7
Enter data:      8

=====Sorting Data=====

                        SORTED DATA IS_____
Data 1 ----> 2
Data 2 ----> 3
Data 3 ----> 4
Data 4 ----> 5
Data 5 ----> 7
Data 6 ----> 8

=====SEARCHING NUMBER=====

Enter data you want to search
4

Middle value is 5
Data found at node 3
```

[Program.codes.cpp@gmail.com.](mailto:Program.codes.cpp@gmail.com)

<https://programcodescpp.wixsite.com/programcodes>

Website:

<https://programcodescpp.wixsite.com/programcodes>

Email

Program.codes.cpp@gmail.com

Facebook Page

<https://web.facebook.com/DJVprogrammers>