Shifting element to left in an array

t

2

|  |  |  |
| --- | --- | --- |
| 0 | 1 | 2 |

|  |  |  |
| --- | --- | --- |
| 2 5 | 5 7 | 7 2 |

**import** **static** java.lang.System.out;

**import** java.util.\*;

**public** **class** Myclass {

**public** **static** **void** main(String[] args) {

Scanner sc=**new** Scanner(System.in);

**int**[] arr= {2, 5, 7};

**int** t=arr[0];//2

1. 1
2. 2

2 3

**int** i;

**for**(i=0;i<arr.length-1 ;i++)

arr[i]=arr[i+1];

//arr[i]=t;//

**for**(**int** n:arr)

out.print(n);//5 7 2

}

}

7

t

Shift element to right in an array

**import** **static** java.lang.System.***out***;

**import** java.util.\*;

|  |  |  |
| --- | --- | --- |
| 0 | 1 | 2 |

|  |  |  |
| --- | --- | --- |
| 2 7 | 5 2 | 7 5 |

**public** **class** Myclass {

**public** **static** **void** main(String[] args) {

Scanner sc=**new** Scanner(System.***in***);

**int**[] arr= {2, 5, 7};

2🡺t

2 1

1 0

**int** t=arr[arr.length-1];

**int** i;

**for**(i=arr.length-1;i>0 ;i--)

arr[i]=arr[i-1];//

arr[i]=t;

1 2 7 8 0

I=1 v=9

1 9 2 7 8

**for**(**int** n:arr)

***out***.print(n); 7 2 5

}

}

TASK🡺 9 5 7 4 3 1 2

Stack

Q🡺Bubble sort

arr

highest element set at the end

Heap

arr

**import** **static** java.lang.System.***out***;

**import** java.util.\*;

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 |
| 0 | 1 | 2 | 3 | 4 |
| 5 | 6 | 2 | 1 | 3 |

**public** **class** Myclass {

**public** **static** **void** main(String[] args) {

8 9 6 5 2 1 3🡺

Scanner sc=**new** Scanner(System.***in***);

I=1

5 2 1 3 6

**2 5** 1 3 6

2 1 5 3 6

2 1 3 **5 6**

**int**[] arr= {5,6,2,1,3};

I=0

b*sort*(arr);

**for**(**int** n:arr)

**5 6** 2 1 3

5 **2 6** 1 3

5 2 **1 6**  3

5 2 1 **3 6**

***out***.print(n);

}

**public** **static** **void** bsort(**int**[] arr) {

**int** t,i,j;

**int** n=arr.length;//5

4

2 1 3 5 6

1 2 3 5 6

**for**(i=0;i<n-1;i++)

4-i(0 1 2 3)

{

**for**(j=0;j<**n-1-i**;j++)

{

**if**(arr[j]>arr[j+1])

5 6//6 2

{ t=arr[j];

1 2 3 5 6

arr[j]=arr[j+1];

arr[j+1]=t;

}

}

}

}

}

8 2 4 3 1

Task🡺 9 5 8 7 3 1 2

Q🡺Selection sort

arr

Lowest element set at the beginning

arr

O index will be compared with all element data

**import** **static** java.lang.System.***out***;

**import** java.util.\*;

Heap

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 |
| 5 | 6  2 | 2 | 1 | 3 |

**public** **class** Myclass {

**public** **static** **void** main(String[] args) {

1

Scanner sc=**new** Scanner(System.***in***);

1 6 5 2 3

1 **5** **6** 2 3

1 **2** 6 **5** 3

5 6 2 1 3

5 6 2 1 3

**2** 6 **5** 1 3

**1** 6 5 2 3

**int**[] arr= {5,6,2,1,3};

s*sort*(arr);

**for**(**int** n:arr)

***out***.print(n);

}

**public** **static** **void** ssort(**int**[] arr) {

**int** t,i,j;

4

1 2 3 6 5

1 2 3 5 6

**int** n=arr.length;//5

1 2 **6 5** 3

1 2 **5 6** 3

1 2 **3** 6 5

**for**(i=0;i<n-1;i++)

3

{

**for**(**j=i+1**;j<**n**;j++)

{

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

{ t=arr[i];

9 6 2 8 3 5 4 1

arr[i]=arr[j];

arr[j]=t;

4 5 5 7 5

4 5 7 5 5 🡺n--

}

}

}

}

}

9 6 2 8 3 5 4 1

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **9** | **6** | 2 | 8 | 3 | 5 | 4 | 1 |
| 6 | 9 | 2 | 8 | 3 | 5 | 4 | 1 |
| **2** | 9 | **6** | 8 | 3 | 5 | 4 | 1 |
| **2** | 9 | 6 | **8** | 3 | 5 | 4 | 1 |
| **2** | 9 | 6 | 8 | **3** | 5 | 4 | 1 |
| **2** | 9 | 6 | 8 | 3 | **5** | 4 | 1 |
| **2** | 9 | 6 | 8 | 3 | 5 | **4** | 1 |
| **2** | 9 | 6 | 8 | 3 | 5 | 4 | **1** |
| **1** | 9 | 6 | 8 | **3** | 5 | 4 | **2** |
|  |  |  |  |  |  |  |  |
| **1** | **9** | **6** | 8 | 3 | 5 | 4 | **2** |
| **1** | **6** | 9 | **8** | 3 | 5 | 4 | 2 |
| **1** | **6** | 9 | 8 | **3** | 5 | 4 | 2 |
| **1** | **3** | 9 | 8 | 6 | **5** | 4 | 2 |
| **1** | **3** | 9 | 8 | 6 | 5 | **4** | 2 |
| **1** | **2** | 9 | 8 | 6 | 5 | 4 | **3** |
|  |  |  |  |  |  |  |  |
| **1** | **2** | 9 | 8 | 6 | 5 | 4 | 3 |
| **1** | **2** | 8 | 9 | 6 | 5 | 4 | 3 |
| **1** | **2** | 6 | 9 | 8 | 5 | 4 | 3 |
| **1** | **2** | **5** | 9 | 8 | **6** | 4 | **3** |
| **1** | **2** | **4** | 9 | 8 | 6 | 5 | **3** |
| **1** | **2** | **3** | 9 | 8 | 6 | 5 | **4** |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |

Q🡺 Accept 5 element in an array , accept a number to be search and print message number found or not? If found also print its index.

import static java.lang.System.out;

import java.util.\*;

arr

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 |
| 5 | 2 | 1 | 8 | 9 |

public class Main {

2000

public static void main(String[] args) {

Scanner sc=new Scanner(System.in);

no

Heap

stack

int[] arr= {5,2,1,8,9};

int no=8;

int index= searchdata(arr,no);

8

if(index!=-1)

out.println("Number Found at index "+index);

else

out.println("Number Not Found");

}

public static int searchdata(int[] arr,int no) {

int i;

for(i=0;i<5;i++)

{ if(arr[i]==no)

{ return i;

}

}

return i;

}

}

public static int searchdata(int[] arr,int no) {

arr

int i,index=-1;

2000

for(i=0;i<5;i++)

{ if(arr[i]==no)

{ index=i;

no

break;

8

}

}

return index;

}

}

Q🡺Binary search

Data has to be in sorted order

It use concept of divide and conquer.

Binary is faster provided data is in sorted order

import static java.lang.System.out;

import java.util.\*;

public class Main {

public static void main(String[] args) {

Scanner sc=new Scanner(System.in);

int no=8;

int[] arr= {1,2,3,4,5,6,7,8,9,10};

bsearch(arr,no);

}

public static void bsearch (int[] arr,int no) {

3

M

U

L

int l=0,u=9,m,flag=0;

0

9

4

while(l<=u) 0 9 0 3 0 <=0 0<=-1

{ m=(l+u)/2; // 0+9/2=4 //0 +3/2= 1 // 0+0/2=0

if(no>arr[m]) //0 [4] 5 0 [1]2 0 [0]1

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |

l=m+1;//

else if(no<arr[m]) // 0 [4]5 0 <[1]2 0 [0]1

u=m-1; // 4-1=3 1-1=0 0-1=-1

Number to be searched **6 3 0 11 5**

6🡺 l=5 u 6 m 5

0🡺 l =0 u-1 m 0

11🡺 l=10 u = 9 m= 9

5🡺 l= u= m=

3🡺l= u= m=

else{

flag=1;

break;

}

} if(flag==1)

out.print("element found");

else

out.print("element Not found");

}

}

N=5

n--;

4🡺0 -3

Accept 5 element in an array having repeated entry , print unique array.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 |
| 2 | 5 | 5 | 5 | 7 |
| 2 | 5 | **5** | **7** | 7 |
| 2 | 5 | 7 | 7 | 7 |

Eg. Input 2 5 5 5 7

O/P= 2 5 7

**import** **static** java.lang.System.***out***;

**import** java.util.\*;

**public** **class** Myclass {

**public** **static** **void** main(String[] args) {

Scanner sc=**new** Scanner(System.***in***);

2 5 6 5 5

**int**[] arr= {2, 5, 5,5,7};

**int** i,j,k;

**int** l=arr.length;//

i=1 j=2

5 5 ==t

k=j(2) l- -(4)

j- - (2-1)🡺1 (j++ 🡺 2)

k=j(2)

**for**(i=0;i<l-1;i++)//1 2 3-1

{

**for**(j=i+1; j<l;j++)// 2 4

K🡺2<5-1(4) 3< (4) 4

{

**if**(arr[i]==arr[j])🡺

{ 3

**for**(k=j;k<l-1;k++)

arr[k]=arr[k+ 1];

j--;// 1

l--;// 3

}

}

}

**for**(**int** n=0;n<l;n++)

***out***.print(arr[n]);

}

}

Accept 5 element in an array and print lowest element and its index

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 |
| 8 | 5 | 3 | 1 | 9 |

o/p🡺1 at index 3

int min=arr[0]; int minindex=0

int size=arr.length

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

{

If(min>arr[i]) //8> 5 5>3 3 >1 1>9==no

{

Min=arr[i]//5 3 1

Minindex=i;//1 2 3

}

}

Min minindex

9 6 4 3 1 2

Public static void selctionsort(int a[],int n)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 |
| 8 | 5 | 3 | 1 | 9 |

Min=i

Min=0

Min=j

Min=1

Min=j

Min=2

Min=j

Min=3

1 5 3 8 9

{ int temp,i,j;

for(i=0;i<n-1;i++)

{ min=i;

for(j=i+1;j<n;j++)

{ if(a[min])>a[j])

min=j;

}

if(min!=i) 3!=0

{ temp=a[min];

a[min]=a[i]

a[i]=temp;

}

}

}

I🡺1 i=2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 |
| 1 | 3 | 5 | 8 | 9 |

Min=i

Min=2 No

No

I=2 Min=2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 |
| 1 | 5 | 3 | 8 | 9 |

Min=i

Min=1

Min=j

Min=2

No

No

I==1 min=2

1 3 5 8 9

I=3

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 |
| 1 | 3 | 5 | 8 | 9 |

Min=I

Min=3 No

I=3 min3

5

9 5 8 6 3 1

1 3 8 6 5 9

Min=i

Min=2

Min=j

Min=3

Min=j

Min=4

1 3 5 6 8 9



1 5 8 6 3 9

Min=i

Min=1

Min=j

Min=4

1 3 8 6 5 9

9 5 8 6 3 1

Min=i

Min=0

Min=j

Min=1

Min=j

Min=4

Min=j

Min=5

1 5 8 6 3 9