

```
➤ ./mergesort
```

```
Enter Array Length
```

```
$ 13
```

```
Before Sorting: 183 86 177 115 193 135 186 92 49 21 162 27 90
```

```
After Sorting: 21 27 49 86 90 92 115 135 162 177 183 186 193
```

```
TIME: 1.7e-05 sec
```

```
➤ ./quicksort
```

```
Enter Array Length
```

```
$ 10
```

```
Before Sorting: 183 86 177 115 193 135 186 92 49 21
```

```
After Sorting: 21 49 86 92 115 135 177 183 186 193
```

```
TIME: 2e-06 sec
```

```
Original Array: 5 9 2 6 11 10 3 21 69 31
```

```
Sorted Array: 2 3 5 6 9 10 11 21 31 69
```

```
Time Taken: 0.000001s
```

```
Original Array: 5 9 2 6 11 10 3 21 69 31
```

```
Sorted Array: 2 3 5 6 9 10 11 21 31 69
```

```
Time Taken: 0.000001s
```

```
Before sorting array elements are -
```

```
47 9 22 42 27 25 0
```

```
After sorting array elements are -
```

```
0 9 22 25 27 42 47
```

```
Enter element to be searched: 2
```

```
Element found!
```

```
Time taken 0.000091 seconds to execute linear search
```

```
Enter element to be searched: 120
```

```
Element found!
```

```
Time taken 0.000074 seconds to execute binary search
```

The Huffman Code:

Data: K, Frequency: 1, Code: 0000
Data: L, Frequency: 1, Code: 0001
Data: E, Frequency: 2, Code: 001
Data: F, Frequency: 4, Code: 01
Data: B, Frequency: 2, Code: 100
Data: C, Frequency: 2, Code: 101
Data: X, Frequency: 2, Code: 110
Data: A, Frequency: 3, Code: 111

Enter number of vertices and edges: 4 5

Enter edges (u v w):

0 1 10

0 2 6

0 3 5

1 3 15

2 3 4

Edges in the Minimum Spanning Tree are:

2 - 3 : 4

0 - 3 : 5

0 - 1 : 10

Total cost of MST = 19

Time taken: 0.000017 seconds

Distance of node1=1

Path=1<-0

Distance of node2=5

Path=2<-3<-0

Distance of node3=3

Path=3<-0

Distance of node4=6

Path=4<-2<-3<-0

Vertex :	0	1	2	3	4
Distance From Source :	0	-1	2	-2	1

Solution for 8 Queens:

```
. Q . . . . .
. . . . Q . . .
. . . . . . Q .
Q . . . . . . .
. . Q . . . . .
. . . . . Q . .
. . . Q . . . .
. . . . . . . Q
```

Time taken: 0.000015 seconds

```
enter the number of row=3
enter the number of column=3
enter the first matrix element=
1 2 3 4 5 6 7 8 9
enter the second matrix element=
9 8 7 6 5 4 3 2 1
multiply of the matrix=
30 24 18
84 69 54
138 114 90
```

Length of LCS is 4

Naive String Matching:

Pattern found at index 10

Rabin-Karp Algorithm:

Pattern found at index 10

KMP Algorithm:

Pattern found at index 10

Total Time taken: 0.000021 seconds

Unsorted Array:

10 30 11 20 4 33 2 1

Sorted Array using Sorting Network:

1 2 4 10 11 20 30 33

Time taken: 0.000018 seconds