

20/6/23

LAB-8

① Heap Sort.

```
void swap (int *a, int *b){  
    int temp = *a;  
    *a = *b;  
    *b = temp;  
}
```

```
void heapify (int arr[], int N, int i){  
    int largest = i;  
    int left = 2*i + 1;  
    int right = 2*i + 2;  
    if (left < N && arr[left] > arr[largest])  
        largest = left;  
    if (right < N && arr[right] > arr[largest])  
        largest = right;  
    if (largest != i){  
        swap (&arr[i], &arr[largest]);  
        heapify (arr, N, largest);  
    }  
}
```

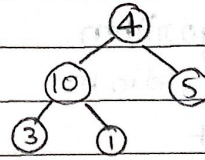
```
void heapSort (int arr[], int N){  
    for (int i = N/2 - 1; i >= 0; i--)  
        heapify (arr, N, i);  
    for (int i = N-1; i > 0; i--){  
        swap (&arr[0], &arr[i]);  
        heapify (arr, i, 0);  
    }  
}
```

```
int main(){  
    printf ("Enter no. of elements");  
    scanf ("%d", &n);  
    printf ("Enter array elements");  
    for (i = 0, i < n; i++)  
        scanf ("%d", &a[i]);  
}
```

3.

O/P:-

Enter no. of elements = 5



1	3	4	5	10
---	---	---	---	----

Enter array elements = 4 10 5 3 1

Sorted array is: 1 3 4 5 10

Time taken = 0.000 sec.

Time taken to sort 500 no. is 0.000

1500 0.015

2500 0.016

3500 0.016

4500 0.047

5500 0.078

6500 0.094

7500 0.124

8500 0.188

9500 0.234

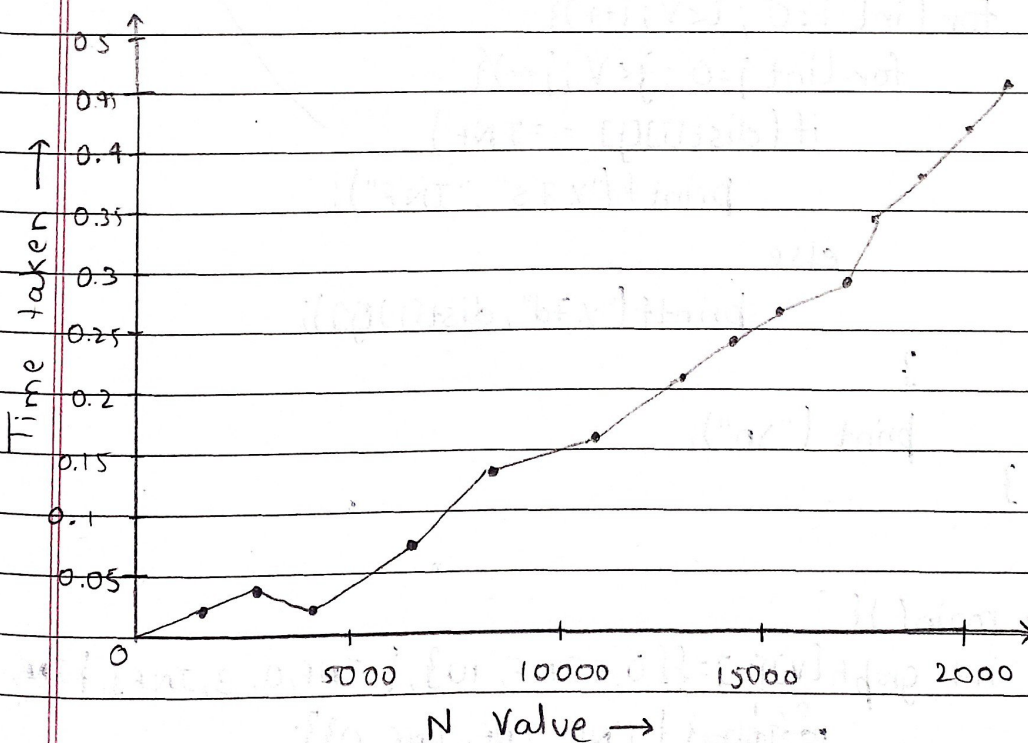
10500 0.266

11500 0.312

12500 0.375

13500 0.407

14500 0.468



② Floyd's Algorithm.

```
#include <stdio.h>
```

```
#define V 4
```

```
#define INF 99999
```

```
void printSolution(int dist[V][V]);
```

```
void FloydWarshall(int dist[V][V]) {
```

```
    int i, j, k;
```

```
    for (k=0; k<V; k++) {
```

```
        for (i=0; i<V; i++) {
```

```
            for (j=0; j<V; j++) {
```

```
                if (dist[i][k] + dist[k][j] < dist[i][j])
```

```
                    dist[i][j] = dist[i][k] + dist[k][j];
```

```
            }
```

```
        }
```

```
    }
```

```
    printSolution(dist);
```

```
}
```

```
void printSolution(int dist[V][V]) {
```

```
    printf("Following matrix shows shortest distance between  
every pair of vertices\n");
```

```
    for (int i=0; i<V; i++) {
```

```
        for (int j=0; j<V; j++) {
```

```
            if (dist[i][j] == INF)
```

```
                printf("%7s", "INF");
```

```
            else
```

```
                printf("%7d", dist[i][j]);
```

```
        }
```

```
        printf("\n");
```

```
    }
```

```
}
```

```
int main() {
```

```
    int graph[V][V] = {{0, 5, INF, 10}, {INF, 0, 3, INF}, {INF, INF,  
0, INF}, {INF, INF, INF, 0}};
```

FloydWarshall (graph);

return 0;

};

O/P: -

Following matrix shows shortest distance b/w every pair of vertices:

0	5	8	9
INF	0	3	4
INF	INF	0	1
INF	INF	INF	0

P
20/6