

20/6/24

DATE: / /

PAGE NO:

Usha Gold

+ Write c programs for :-

1) Heap sort

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

```
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 (largest != i) {
```

```
        swap (&arr[i], &arr[largest]);
```

```
        heapify (arr, N, largest);
```

```
    }
```

```
}
```



```

void printarray (int arr[], int N)
{
    for (int i = 0; i < N; i++)
        printf ("%d", arr[i]);
    printf ("\n");
}

```

```

int main ()
{
    int arr[] = { 5, 35, 25, 45, 30, 55, 25, 45 };
    int N = sizeof (arr) / sizeof (arr[0]);

    heapsort (arr, N);
    printf ("Sorted array is \n");
    printarray (arr, N);
}

```

Output:-

Enter the no of elements :- 8

Time Taken to sort 8 no of elements is 0.000121 sec

Sorted array is

5, 10, 25, 30, 35, 45, 50, 55

Time complexity for N no of elements:

5000 is 0.015000 sec

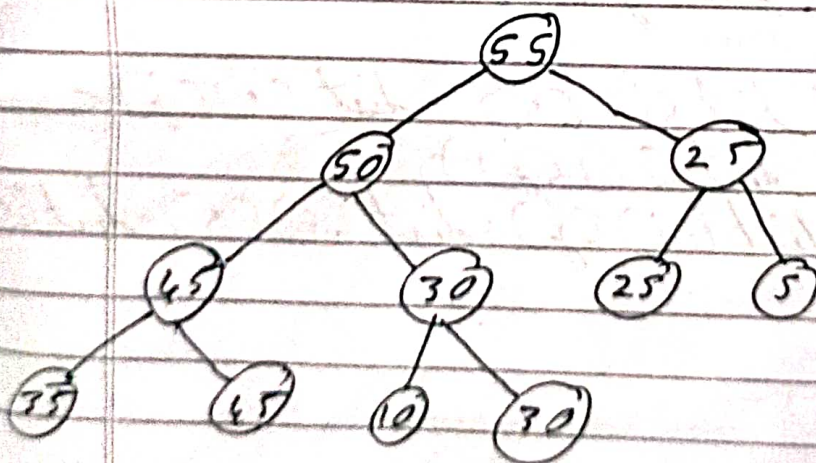
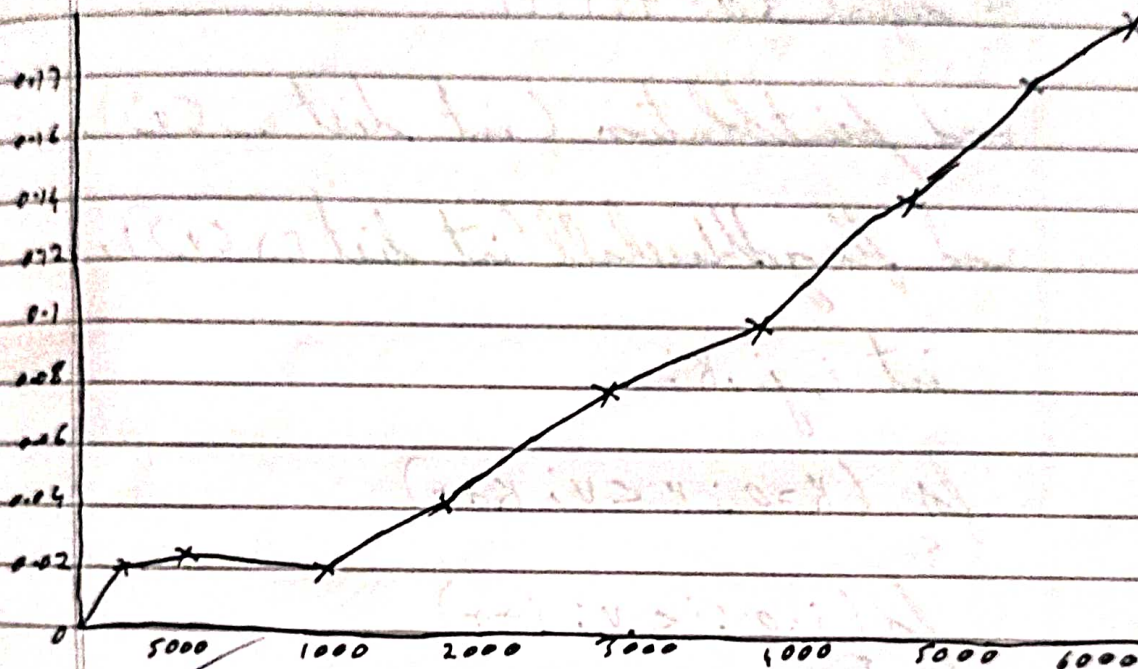
6000 is 0.016000 sec

7000 is 0.031 sec

8500 is 0.071 sec



8000 is 1.72 sec  
8500 is 0.46 sec  
9000 is 0.067 sec  
9500 is 0.079 sec  
10000 is 0.093 sec





+ shortest path problem using Floyd's algo.

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

```
#define V4
```

```
#define INF 999999
```

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

```
void floydWarshall (int dist [][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];
```

```
  }
```

```
}
```

```
}
```

```
void Pl (int dist [][V])
```

```
{
```

```
  printf (
```



```

for(int i=0; i<m; i++)
{
    for(int j=0; j<v; j++) {
        if (dist[i][j] == INF)
            printf("%7d", "INF");
        else
            printf("%7d", dist[i][j]);
    }
    printf("\n");
}
}

```

```

int main()
{

```

```

    int graph[v][v] = { { 0, 4, N, 5, N },
                        { 5, 0, 1, 5, 6 },
                        { 2, N, 0, 3, N },
                        { N, N, 1, 0, 2 },
                        { 1, N, N, 4, 0 } }

```

W. 20/6/24

```

    FloydWarshall(graph);
    return 0;
}

```

Output:-

The following matrix shows the shortest distance between every pair of vertices

0	4	5	5	7
3	0	1	4	6
2	6	0	3	5
3	7	1	0	2
1	5	5	4	0