CSA0389

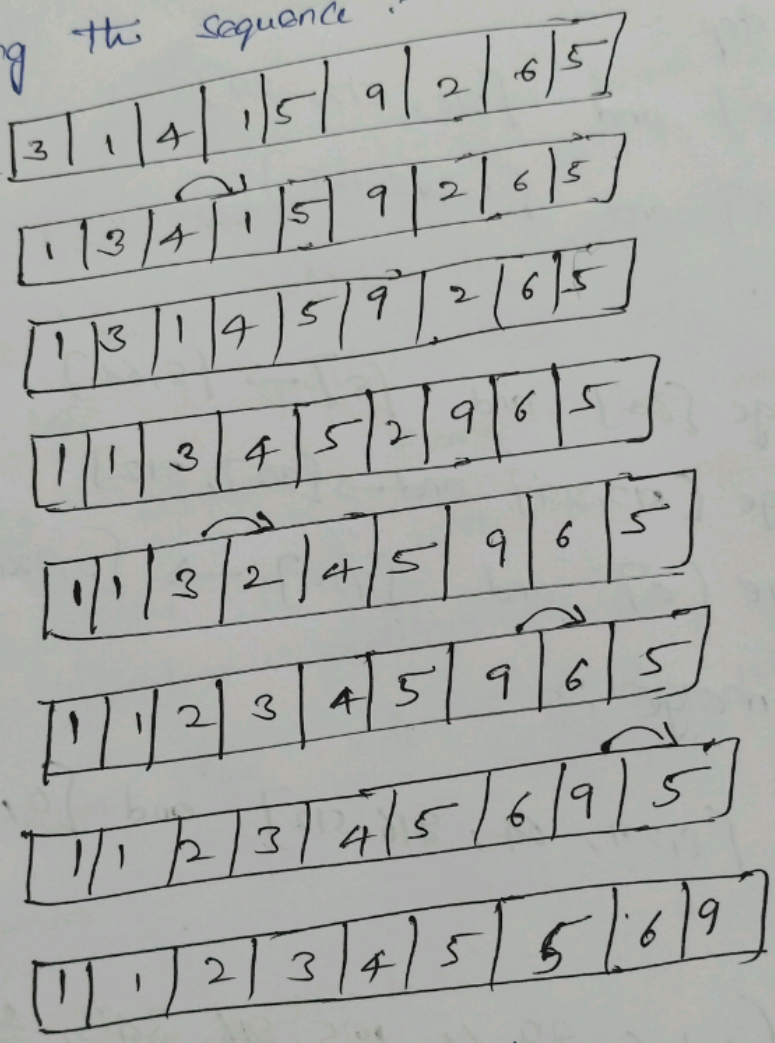
K. Nithesh
192325090

Assignment - 5

1. Illustrate the queue operation using following function calls of size = 5. following sequence 3, 1, 4, 5, 9, 2, 6, 5. Explain the procedure for merge sort and perform merge sort each step of iteration 64, 8, 216, 512, 72, 90, 343, 125.

(SOL)

Sorting the sequence :-



Merge sort procedure :-

* Split the list into halves until each sublist has one element.
* Combine the sublist to procedure new sorted sublists until there is one sorted list.

Merge sort with 64, 8, 216, 512, 27, 729, 0, 1, 343, 125

Initial step :

[64, 8, 216, 512, 27] and [729, 0, 1, 343, 125]

further step :-

[64, 8] and [216, 512, 27]

[729, 0] and [1, 343, 125]

Merge :-

Merge [64] and [8] → [8, 64]

Merge [512, 27] and → [27], 512]

Merge [0] and [729] → [0, 729]

Final merge :-

Merge [8, 27, 64, 216, 512] and [0, 1, 125, 343, 729]

→ [0, 1, 8, 27, 64, 125, 216, 343, 512, 729]

2. Draw the concept map of the following in quick sort try to write on algorithm for it, which is as follows & develop a program considers.

Sol

Steps :-

* Select the element at the highest index as the pivot.
* set 'left' to the low index & right to the high index.
* move 'left' right words and right leftwards until left is greator than (or) equal to 'right' swapping elements as needed
* Swap the pivot with the element at the left pointer position
* return the index of the pivot.

Program :-

```
#include <stdio.h>
int main (){
int arr[] = {64, 8, 216, 512, 27, 729};
int n = size of (arr) / size of (arr(0))
int low = 0; high = n-1;
while (low < high) {
```

```
int pivot = arr[high];
int left = low;
while (right >= low && arr[right] > pivot){
if (left < right){
    int temp = arr[left];
    arr[left] = arr[right];
    arr[right] = temp;
    left++
print ("sorted array");
  for (int i=0; i<n; i++);
    printf("%d", arr[i]);
  }
    printf("\n");
    return 0;
}
```

output :-

0, 1, 8, 27, 64, 125, 216, 343, 512, 729