EB3/67264/23

ELOSY MWENDWA

COSC 333

The Implementation: Quicksort in C

Strategy: **Divide & Conquer**

Speed: ***O (n log n)***

```
#include <stdio.h>


// Function to swap two elements
void swap(int* a, int* b) {
   int t = *a;
   *a = *b;
   *b = t;
}
int partition(int arr[], int low, int high) {
   int pivot = arr[high];
   int i = (low - 1);
   int j;


   for (j = low; j <= high - 1; j++) {
      if (arr[j] < pivot) {
         i++;
         swap(&arr[i], &arr[j]);
      }
   }
   swap(&arr[i + 1], &arr[high]);
   return (i + 1);
}
```

```
// The Quicksort function
void quickSort(int arr[], int low, int high) {
   if (low < high) {
      int pi = partition(arr, low, high);
      quickSort(arr, low, pi - 1);
      quickSort(arr, pi + 1, high);
   }
}
int main() {
   int n;
   int i;
   printf("Enter the number of elements: ");
   scanf("%d", &n);

   int arr[n];
   printf("Enter %d integers:\n", n);
   for (i = 0; i < n; i++) {
      scanf("%d", &arr[i]);
   }
   quickSort(arr, 0, n - 1);

   printf("\nSorted array: \n");
   for (i = 0; i < n; i++) {
      printf("%d ", arr[i]);
   }
   printf("\n");
   return 0;
}
```