#include <iostream>  
using namespace std;  
  
int getMax(int arr[], int n) {  
 int mx = arr[0];  
 for (int i = 1; i < n; i++)  
 if (arr[i] > mx)  
 mx = arr[i];  
 return mx;  
}  
  
void countSort(int arr[], int n, int exp) {  
 int output[n];  
 int i, count[10] = {0};  
  
 for (i = 0; i < n; i++)  
 count[(arr[i] / exp) % 10]++;  
  
 for (i = 1; i < 10; i++)  
 count[i] += count[i - 1];  
  
 for (i = n - 1; i >= 0; i--) {  
 output[count[(arr[i] / exp) % 10] - 1] = arr[i];  
 count[(arr[i] / exp) % 10]--;  
 }  
  
 for (i = 0; i < n; i++)  
 arr[i] = output[i];  
}  
  
void radixsort(int arr[], int n) {  
 int m = getMax(arr, n);  
  
 for (int exp = 1; m / exp > 0; exp \*= 10)  
 countSort(arr, n, exp);  
}  
  
void print(int arr[], int n) {  
 for (int i = 0; i < n; i++)  
 cout << arr[i] << " ";  
}  
  
int main() {  
 int arr[] = {380, 23, 75, 18, 744, 57, 4, 39};  
 int n = sizeof(arr) / sizeof(arr[0]);  
  
 radixsort(arr, n);  
 print(arr, n);  
 return 0;  
}

The time complexity of radix sort is given by the formula, T(n) = O (d\*(n + b)), where d is the number of digits in the given list, n is the number of elements in the list, and b is the base or bucket size used, which is normally base 10 for decimal representation, for example, for the decimal system, b is 10.