Shreyansh – 17csu184

**RADIX SORT**

**CODE**

#include<stdio.h>

#include<time.h>

int get\_max (int a[], int n){

int max = a[0];

for (int i = 1; i < n; i++)

if (a[i] > max)

max = a[i];

return max;

}

void radix\_sort (int a[], int n){

int bucket[10][10], bucket\_cnt[10];

int i, j, k, r, NOP = 0, divisor = 1, lar, pass;

lar = get\_max (a, n);

while (lar > 0){

NOP++;

lar /= 10;

}

for (pass = 0; pass < NOP; pass++){

for (i = 0; i < 10; i++){

bucket\_cnt[i] = 0;

}

for (i = 0; i < n; i++){

r = (a[i] / divisor) % 10;

bucket[r][bucket\_cnt[r]] = a[i];

bucket\_cnt[r] += 1;

}

i = 0;

for (k = 0; k < 10; k++){

for (j = 0; j < bucket\_cnt[k]; j++){

a[i] = bucket[k][j];

i++;

}

}

divisor \*= 10;

printf ("After pass %d : ", pass + 1);

for (i = 0; i < n; i++)

printf ("%d ", a[i]);

printf ("\n");

}

}

int main (){

clock\_t start;

int i, n, a[10];

int total;

printf ("Enter the number of items to be sorted: ");

scanf ("%d", &n);

printf ("Enter items: ");

for (i = 0; i < n; i++){

scanf ("%d", &a[i]);

}

radix\_sort (a, n);

printf ("Sorted items : ");

for (i = 0; i < n; i++)

printf ("%d ", a[i]);

printf ("\n");

clock\_t end;

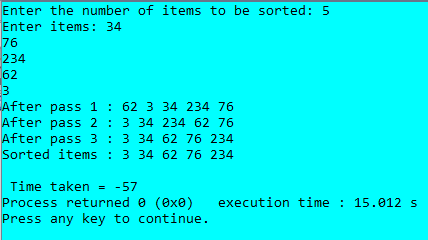
total= end - start;

printf("\n Time taken = %d",total);

return 0;

}

**OUTPUT:**



**HEAP SORT**

**CODE**

#include <stdio.h>

#include<time.h>

int main(void)

{ clock\_t start;

int total;

int heap[10], no, i, j, c, root, temp;

printf("\nEnter no of elements :");

scanf("%d", &no);

printf("\nEnter the numbers : ");

for (i = 0; i < no; i++)

scanf("%d", &heap[i]);

for (i = 1; i < no; i++)

{

c = i;

do

{

root = (c - 1) / 2;

if (heap[root] < heap[c])

{

temp = heap[root];

heap[root] = heap[c];

heap[c] = temp;

}

c = root;

} while (c != 0);

}

printf("enter elements : ");

for (i = 0; i < no; i++)

printf("%d\t ", heap[i]);

for (j = no - 1; j >= 0; j--)

{

temp = heap[0];

heap[0] = heap[j];

heap[j] = temp;

root = 0;

do

{

c = 2 \* root + 1;

if ((heap[c] < heap[c + 1]) && c < j-1)

c++;

if (heap[root]<heap[c] && c<j)

{

temp = heap[root];

heap[root] = heap[c];

heap[c] = temp;

}

root = c;

} while (c < j);

}

printf("\n sorted array is : ");

for (i = 0; i < no; i++)

printf("\t %d", heap[i]);

clock\_t end;

total=end-start;

printf("\n total time taken is %d", total);

return 0;

}

**OUTPUT:**

