Counting Sort

Counting sort is a sorting technique based on keys between a specific range. It works by counting the number of objects having distinct key values (kind of hashing). Then doing some arithmetic to calculate the position of each object in the output sequence.

Applications

Counting sort is used when:

- there are smaller integers with multiple counts.
- linear complexity is the need.

C program to implement counting sort

```
#include <stdio.h>

#include<conio.h>

 void counting_sort(int A[], int k, int n)

{

   int i, j;

   int B[15], C[100];

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

      C[i] = 0;

   for (j = 1; j <= n; j++)

      C[A[j]] = C[A[j]] + 1;

   for (i = 1; i <= k; i++)

      C[i] = C[i] + C[i-1];
```

```
    for (j = n; j >= 1; j--)
    {
        B[C[A[j]]] = A[j];
        C[A[j]] = C[A[j]] - 1;
    }
    printf("The Sorted array is : ");
    for (i = 1; i <= n; i++)
        printf("%d ", B[i]);
}
void main()
{
    int n, k = 0, A[15], i;
    printf("Enter the number of input : ");
    scanf("%d", &n);
    printf("\nEnter the elements to be sorted :\n");
    for (i = 1; i <= n; i++)
    {
        scanf("%d", &A[i]);
        if (A[i] > k) {
            k = A[i];
        }
```

```
    }
    counting_sort(A, k, n);
    printf("\n");
    getch();
}
```