Name: Muhammad Uzair

Roll No. L1F18BSCS0311

Section - F



**As if value of k is not given for the counting sort algorithm, how will you get the value of k from array A. Write what is added to this current available code.**

**Answer**

COUNTING-SORT(A,B,K)

Let k = A[0]

for i = 0 to A.length

if(A[i] > k)

k = A[i]

k = k + 1

Let C[0..k] be a new array

Rest is the above given code.

**As general range of keys is [0 to k], but if the minimum key value is not 0 and rather it is much greater, will you change the counting sort code? With reason write your answer.**

**Answer**

Yes I will change the counting sort code because, in counting sort we have keys [0 to k], so, if the minimum key is much greater, and the numbers in the array are less than the greater minimum key value, we won't be able to count the numbers and put the count of numbers in their assigned index place.

**As general range of keys is [0 to k], but if now key values are –ve, 0 and +ve, how will you change the counting sort code? With reason write your answer in detail and if needed write the new counting sort pseudo code.**

**Answer**

As the keys in counting sort code range from [0 to k], but when the key values are -ve, 0, +ve, so we would have to find the minimum and maximum element in the array. Minimum will be multiplied by 1 to make it positive. Then we will create 2 arrays, one for positive numbers of size max and one for negative numbers of size min. Then we will initialize array of +ve and array of -ve numbers with 0. Then we will store frequencies of occurences of +ve and -ve in their respective arrays. Then we will perform running sum on array of -ve numbers from min to 0. Then we will perform running sum on array of +ve numbers from 0 to max. Then we will store the final result in out output array. Following is the changed counting sort code:

void counting-sort(int A[], int B[])

{

int max = A[0], min = A[0];

for(int i = 0; i < A.length; i++)

{

if(A[i] > max)

{

max = A[i];

}

}

for(int i = 0; i < A.length; i++)

{

if(A[i] < min)

{

min = A[i];

}

}

min \*= -1;

int P[max], N[min];

for(int i = 1; i < A.length; i++)

{

if(A[i] >= 0)

{

P[A[i]] = P[A[i]] + 1;

}

else if(A[i] < 0)

{

N[A[i]\* - 1] = N[A[i] \* -1] + 1;

}

}

for(int i = min - 1; i >= 0; i--)

{

N[i] = N[i] + N[i+1];

}

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

{

P[i] = P[i] + P[i - 1];

}

for(int i = 0; i <= max; i++)

{

P[i] = P[i] + N[0];

}

for(int i = 1; i < A.length; i++)

{

if(A[i] >= 0)

{

B[P[A[i]]] = A[i];

P[A[i]] = P[A[i]] - 1;

}

else if(A[i] < 0)

{

B[N[A[i] \* -1]] = A[i];

N[A[i] \* -1] = N[A[i] \* -1] - 1;

}

}

}

**Apply radix sort on the given array of values show all passes on the given array. [keep I mind the ASCII values], all 5 steps  
Arr = {10124, 00256, 11005, 10205, 90125, 32150, 23154, 20014, 00325, 65002, 14005, 89201, 74005, 56231, 85002, 65214}**

**Step 1**

Arr = {32150, 89201, 56231, 65002, 85002, 10124, 23154, 20014, 65214, 11005, 10205, 90125, 00325, 14005, 74005, 00256}

**Step 2**

Arr = {89201, 65002, 85002, 11005, 10205, 14005, 74005, 20014, 65214, 10124, 90125, 00325, 56231, 32150, 23154, 00256}

**Step 3**

Arr = {65002, 85002, 11005, 14005, 74005, 20014, 10124, 90125, 32150, 23154, 89201, 10205, 65214, 56231, 00256, 00325}

**Step 4**

Arr = {20014, 10124, 90125, 10205, 00256, 00325, 11005, 32150, 23154, 14005, 74005, 65002, 85002, 65214, 56231, 89201}

**Step 5**

Arr = {00256, 00325, 10124, 10205, 11005, 14005, 20014, 23154, 32150, 56231, 65002, 65214, 74005, 85002, 89201, 90125}