Sort n numbers in range from 0 to n^2 – 1 in linear time

Given an array of numbers of size n. It is also given that the array elements are in range from 0 to $n^2 – 1$. Sort the given array in linear time.

```
Examples:
Since there are 5 elements, the elements can be from 0 to 24.
Input: arr[] = {0, 23, 14, 12, 9}
Output: arr[] = {0, 9, 12, 14, 23}

Since there are 3 elements, the elements can be from 0 to 8.
Input: arr[] = {7, 0, 2}
Output: arr[] = {0, 2, 7}
```

We strongly recommend to minimize the browser and try this yourself first.

Solution: If we use Counting Sort, it would take O(n^2) time as the given range is of size n^2. Using any comparison based sorting like Merge Sort, Heap Sort, .. etc would take O(nLogn) time.
Now question arises how to do this in 0(n)? Firstly, is it possible? Can we use data given in question? n numbers in range from 0 to $n^2 - 1$?
The idea is to use Radix Sort. Following is standard Radix Sort algorithm.

```
1) Do following for each digit i where i varies from least
   significant digit to the most significant digit.
……..a) Sort input array using counting sort (or any stable
        sort) according to the i'th digit
```

Let there be d digits in input integers. Radix Sort takes O(d*(n+b)) time where b is the base for representing numbers, for example, for decimal system, b is 10. Since n^2-1 is the maximum possible value, the value of d would be [Tex]O(\log_b(n))[/Tex]. So overall time complexity is [Tex]O((n+b)*\log_b(n))[/Tex]. Which looks more than the time complexity of comparison based sorting algorithms for a large k. The idea is to change base b. If we set b as n, the value of [Tex]O(\log_b(n))[/Tex] becomes O(1) and overall time complexity becomes O(n).

```
arr[] = {0, 10, 13, 12, 7}

Let us consider the elements in base 5. For example 13 in
base 5 is 23, and 7 in base 5 is 12.
arr[] = {00(0), 20(10), 23(13), 22(12), 12(7)}

After first iteration (Sorting according to the last digit in
base 5),  we get.
arr[] = {00(0), 20(10), 12(7), 22(12), 23(13)}

After second iteration, we get
arr[] = {00(0), 12(7), 20(10), 22(12), 23(13)}
```

Following is C++ implementation to sort an array of size n where elements are in range from 0 to $n^2 - 1$.

```
#include<iostream>
using namespace std;

// A function to do counting sort of arr[] according to
// the digit represented by exp.
int countSort(int arr[], int n, int exp)
{

    int output[n]; // output array
    int i, count[n] ;
    for (int i=0; i < n; i++)
       count[i] = 0;

    // Store count of occurrences in count[]
    for (i = 0; i < n; i++)
        count[ (arr[i]/exp)%n ]++;

    // Change count[i] so that count[i] now contains actual
    // position of this digit in output[]
    for (i = 1; i < n; i++)
        count[i] += count[i - 1];
```

```
    // Build the output array
    for (i = n - 1; i >= 0; i--)
    {
        output[count[ (arr[i]/exp)%n] - 1] = arr[i];
        count[(arr[i]/exp)%n]--;
    }

    // Copy the output array to arr[], so that arr[] now
    // contains sorted numbers according to curent digit
    for (i = 0; i < n; i++)
        arr[i] = output[i];
}


// The main function to that sorts arr[] of size n using Radix Sort
void sort(int arr[], int n)
{
    // Do counting sort for first digit in base n. Note that
    // instead of passing digit number, exp (n^0 = 0) is passed.
    countSort(arr, n, 1);

    // Do counting sort for second digit in base n. Note that
    // instead of passing digit number, exp (n^1 = n) is passed.
    countSort(arr, n, n);
}

// A utility function to print an array
void printArr(int arr[], int n)
{
    for (int i = 0; i < n; i++)
        cout << arr[i] << " ";
}

// Driver program to test above functions
int main()
{
    // Since array size is 7, elements should be from 0 to 48
    int arr[] = {40, 12, 45, 32, 33, 1, 22};
    int n = sizeof(arr)/sizeof(arr[0]);
    cout << "Given array is \n";
    printArr(arr, n);

    sort(arr, n);

    cout << "\nSorted array is \n";
    printArr(arr, n);
    return 0;
}
```

Output:

```
Given array is
40 12 45 32 33 1 22
```

```
Sorted array is
1 12 22 32 33 40 45
```

How to sort if range is from 1 to n^2?

If range is from 1 to n n^2, the above process can not be directly applied, it must be changed. Consider n = 100 and range from 1 to 10000. Since the base is 100, a digit must be from 0 to 99 and there should be 2 digits in the numbers. But the number 10000 has more than 2 digits. So to sort numbers in a range from 1 to n^2, we can use following process.

1) Subtract all numbers by 1.

2) Since the range is now 0 to n^2, do counting sort twice as done in the above implementation.

3) After the elements are sorted, add 1 to all numbers to obtain the original numbers.

How to sort if range is from 0 to n^3 -1?

Since there can be 3 digits in base n, we need to call counting sort 3 times.

This article is contributed by **Bateesh**. Please write comments if you find anything incorrect, or you want to share more information about the topic discussed above

Related Topics:

- Find Union and Intersection of two unsorted arrays
- Pythagorean Triplet in an array
- Maximum profit by buying and selling a share at most twice
- Design a data structure that supports insert, delete, search and getRandom in constant time
- Print missing elements that lie in range 0 – 99
- Iterative Merge Sort
- Group multiple occurrence of array elements ordered by first occurrence
- Given a sorted and rotated array, find if there is a pair with a given sum

Writing code in comment? Please use **ideone.com** and share the link here.

dodo • 25 days ago

Hi, cant we create bst and then do inorder in that..??

Anurag Singh ➚ dodo • 16 hours ago

We can. But bst creation with n nodes would take O(n Log n) (if balanced), so overall complexity won't be linear

• Reply • Share ›

Mustafa • a month ago

Can anyone explain how does counting sort take O(n^2) time..

• Reply • Share ›

prem singh Mustafa • 13 days ago

In the Counting sort we create array of size of range of number and map all the values to array. So here we have to create a array of size n^2 and map all the n values to the array.
You can read more about counting sort here:
http://www.geeksforgeeks.org/c...

• Reply • Share ›

Bhavik • 5 months ago

Total Number of digits would be log base b(n^2-1) rather than log base b(n).

• Reply • Share ›

neharika Bhavik • 4 months ago

yes you r ryt. But in the above explanation, O(log base b(n)) is written and not log base b(n).

• Reply • Share ›

Vikas Malviya • 9 months ago

How to sort if range is from 0 to n^3 -1?
Since there can be 3 digits in base n, we need to call counting sort 3 times.

So, what should be the values of exp for the three cases??

• Reply • Share ›

Sumit Kesarwani Vikas Malviya • 3 months ago

See the radix sort... yu will get it..becz what ever u r asking this is part of the radix sort...
please correct me if i am wrong..

• Reply • Share ›

ultimate_coder Vikas Malviya • 9 months ago

1,n,n^2

1 • Reply • Share ›

DS+Algo=Placement • 9 months ago

Plz answer someone,
In range 1 to n^2, what is bothering 1 or n^2 or both

• Reply • Share ›

Terry • 10 months ago

Why algorthm have O(n) complexity, but (arr[i]/exp)%n isn't O(1)?

• Reply • Share ›

typing.. • 10 months ago

m little bit confused here, if n=10000, and range is n^3-1 , then it require calling counting sort 4 times, so, why is it written here:

since there can be 3 digits in base n, we need to call counting sort 3 times in case of n^3-1

• Reply • Share ›

GOPI GOPINATH ➚ typing.. • 10 months ago

When we say range is from 1 to n^3 (ex:n=10) then range is from 1 to 1000. So we modify it to 0 to 999. Hence we need to apply count sort trice.

• Reply • Share ›

Vishal Shaw • a year ago

suppose that the problem is to sort n^k-1 range of numbers then we need to call counting sort k times but what parameters will be passed for these calling functions?

• Reply • Share ›

typing.. ➚ Vishal Shaw • 10 months ago

No Vishal, we have to pass n^i (where i is from 0 onwards) until we found (m/i)>0 where m is largest number among all of array elements.

• Reply • Share ›

Vishal Shaw ➚ typing.. • 10 months ago

but then it is simply radix sort... isn't it? but what i want 2 know how we modify the given code to generalize for any k?

• Reply • Share ›

typing.. ➚ Vishal Shaw • 10 months ago

I understand what you want to convey,you are correct, thats why I posted the confusion again.

• Reply • Share ›

Utkarsh Dubey • a year ago

"Note that instead of passing digit number, exp (n^0 = 0) is passed."
Please correct this to exp(n^0 = 1). It has gone unnoticed pretty long. Also could someone explain why we send the exponents instead of digits?

4 • Reply • Share ›

Ajit Kumar • a year ago

You could use bucket sort. For each number k in array put it in bucket sqroot(k). If numbers are uniformly distributed, sorting will take O(n) similar to http://www.geeksforgeeks.org/b...

• Reply • Share ›

GOPI GOPINATH • a year ago

I could think of a solution. if the range of numbers => n^2-1 is given..then we can simply create a auxiliary array with sorted numbers from 0 to n^2-1 and traverse through the actual array and mark the elements in the auxiliary array. Now traverse the auxiliary array and print if the number is marked. (we may copy back to actual array) . Time complexity O(n) and space complexity O(n^2)
But linear time complexity is achieved :P

• Reply • Share ›

dodo ➚ GOPI GOPINATH • 25 days ago

hi, cant we create a bst and do inorder in that.?? while doing in order i will put the elements in sorted order inside the array.

• Reply • Share ›

GOPI GOPINATH ➚ dodo • 25 days ago

you won't get linear complexity but O(n log n)

• Reply • Share ›

GOPI GOPINATH ➚ GOPI GOPINATH • a year ago

Time complexity is also O(n^2) as we need to traverse all the n^2-1 elements. so my approach doesnt suit. if the range is 'n' then it would do.

• Reply • Share ›

tweety ➚ GOPI GOPINATH • 10 months ago

nice approach... but i think if range is n still time complexity would be O(n^2). because for each element in the array we need to traverse auxiliary array....

• Reply • Share ›

GOPI GOPINATH ➚ tweety • 10 months ago

Yes u r ryt. I ve mentioned the same in the previous comment :-)

• Reply • Share ›

tweety ➚ GOPI GOPINATH • 10 months ago

sry if m wrong but u have said if range is n then it would do.... but i think that it doesnt depend on range.... complexity would be O(n^2) always....

• Reply • Share ›

GOPI GOPINATH ➚ tweety • 10 months ago

If the range is 0 to n then just create auxiliary array of size n. Mark those

numbers appearing in the original array in auxiliary array.(Increment counter , which ll help if duplicate elements are present). Now traverse the Auxiliary array and print those values which are marked (or having counter more than 0). overall complexity is O(2n)=>O(n) .

Can u please comment How this becomes O(n^2) ?

• Reply • Share ›

tweety ➚ GOPI GOPINATH • 10 months ago

o i got it.... u r right... i was thinking something else... you are using concept of counting sort.... thankxx.. :)

• Reply • Share ›

Riju ➚ GOPI GOPINATH • a year ago

Whats the significance of n^2-1 then? Any range would do.

• Reply • Share ›

GOPI GOPINATH ➚ Riju • a year ago

we dn knw the upper limit. Here we knw the upper and lower limits.

• Reply • Share ›

satinjal • a year ago

may u please refine the comment within sort function definition -> "instead of passing digit number, exp (n^0 = 0) is passed". It would be more easier for beginner.

• Reply • Share ›

sw • a year ago

The math doesnt seem right. You can replace one constant with the other while changing the bases in log. What if n is around 100 billion? The point is n is not constant so cant be used to replace a constant.

1 • Reply • Share ›

Hitesh • a year ago

Nice post! Building the output array cane be done in forward direction loop after making required changes in count loop. That would make the life simpler!

• Reply • Share ›

zzer ➚ Hitesh • a year ago

actually, we have to iterate backward to keep the sort stable.

• Reply • Share ›

Srithar • a year ago

This is called Radix Sort. No need for an article.

1 • Reply • Share ›

guest ➚ Srithar • a year ago

It actually uses radix sort as a subroutine..Lot of questions are there which internaly uses some sorting algorithm...Lot of people cant relate this to radix sort..We need it as a separate post.:)

7 • Reply • Share ›

guest ➚ guest • a year ago

this is radix sort article true, also question is in excersie of chapter "Sorting in linear time " in coremen. But a good reminder.

• Reply • Share ›

sonu • a year ago

This question can be easily solved if Space is not important . We can use extra space of size N^2 - 1 using bit vector / Array (B) . Sol for (i = 0 ; i < length ; i++) B[a[i]] = 1 ;

for(i = 0 ; i < n*n ; i++) if (B[i] = = 1) cout << i <<endl ;="">

• Reply • Share ›

Dev ➚ sonu • a year ago

What about repeating elements? Also you are iterating a loop with n^2 elements and hence n^2 complexity!

1 • Reply • Share ›

veer • a year ago

this problem can be solved in few loops.
b[] contain array of number to be sorted

first create a array of size =number^2 let a[]

then for every element in 'a' assign zero

same do for r[] where r[] is to keep track of number of repetition of number.
now assign 1 to the particular element in a[] who's index is b[i].
now reassign to input array by checking if element of a[i]is equal to 1 then assign index ob a[] to b[j]=i.
while take care of repetition by using value in r[i].

this algo will be O(n) if no repetition other wise ~=O(n)

here is sample code (
sorry it is not displaYing properly)

// x is n^2
for(i=0;i<x;i++){ a[i]="0;" r[i]="0;" }="" for(i="0;i<number;i++){" a[b[i]]="1;" r[b[i]]="r[b[i]]+1;" if=""

for(i=0;i<x;i++){ a[i]=0; r[i]=0; } for(i=0;i<number;i++){ a[b[i]]=1; r[b[i]]=r[b[i]]+1; } i=
array="" contain="" repeated="" number="" }="" j="0;" for(i="0;i<x;i++){" if(a[i]="=1){" b[j]="i;"
j++;="" }="" }="" cout<<"\n";="" for(i="0;i<number;i++){" for(j="0;j<r[b[i]];j++){" cout<<b[i]
<<"="" ";="" }="">

• Reply • Share ›

Abhishek ➚ veer • a year ago

Yes you are right but you are using more space and logically you are making time complexity to O(n2) because you are going from i=0 to i < n2 in one loop which logically equal to run the two nested loop of size n.

• Reply • Share ›

veer ➚ Abhishek • a year ago

ya ,bro u r right.i didnt notice. sorry

• Reply • Share ›

Abhishek • a year ago

I think math behind the problem and the coding solution are not same because when we call countSort(arr, n, 1) first time it is sorting array according to the range of remainder (Mean if N = 5 than range or remainder is 0 to 4 and it is clustering all the numbers in the assending order of remainder and in second call it is sorting the number on the basics of divison means if we have two numbers 22 and 24 than dividend is 5 for both but reminder is 2 and 4 so algorithm will arrange them as 22 and 24).

Correct Me If I am going on the wrong track

• Reply • Share ›

Kartik ➚ Abhishek • a year ago

Abhishek, please note that the counting sort used in the program is a stable sorting algorith, you can verify the theory and code by printing intermediate count[] and arr[] values.

• Reply • Share ›

Name • a year ago

ingenious*

• Reply • Share ›

joker ➚ Name • 6 months ago

yeah actually that is the spelling of genius, so its quite confusing.

• Reply • Share ›

andy qi • a year ago

ingenius!

3 • Reply • Share ›

- Understanding “extern” keyword in C
- Median of two sorted arrays
- Tree traversal without recursion and without stack!
- Structure Member Alignment, Padding and Data Packing
- Intersection point of two Linked Lists
- Lowest Common Ancestor in a BST.
- Check if a binary tree is BST or not
- Sorted Linked List to Balanced BST

- Follow @GeeksforGeeks

Recent Comments

- lt_k

  i need help for coding this function in java...

  Java Programming Language · 1 hour ago

- Piyush

  What is the purpose of else if (recStack[*i])...

  Detect Cycle in a Directed Graph · 1 hour ago

- Andy Toh

  My compile-time solution, which agrees with the...

  Dynamic Programming | Set 16 (Floyd Warshall Algorithm) · 1 hour ago

- lucy

  because we first fill zero in first col and...

  Dynamic Programming | Set 29 (Longest Common Substring) · 2 hours ago

- lucy

  @GeeksforGeeks i don't n know what is this long...

  Dynamic Programming | Set 28 (Minimum insertions to form a palindrome) · 2 hours ago

- manish

  Because TAN is not a subsequence of RANT. ANT...

  Given two strings, find if first string is a subsequence of second · 2 hours ago