GeeksQuiz

Computer science mock tests for geeks

Selection Sort

The selection sort algorithm sorts an array by repeatedly finding the minimum element (considering ascending order) from unsorted part and putting it at the beginning. The algorithm maintains two subarrays in a given array.

1) The subarray which is already sorted.
2) Remaining subarray which is unsorted.

In every iteration of selection sort, the minimum element (considering ascending order) from the unsorted subarray is picked and moved to the sorted subarray.

Following example explains the above steps:

```
arr[] = 64 25 12 22 11

// Find the minimum element in arr[0...4] and place it at beginning
11 25 12 22 64

// Find the minimum element in arr[1...4] and
// place it at beginning of arr[1...4]
11 12 25 22 64

// Find the minimum element in arr[2...4] and
// place it at beginning of arr[2...4]
11 12 22 25 64

// Find the minimum element in arr[3...4] and
// place it at beginning of arr[3...4]
11 12 22 25 64
```

```
// C program for implementation of selection sort
#include <stdio.h>
```

```
void swap(int *xp, int *yp)
{
    int temp = *xp;
    *xp = *yp;
    *yp = temp;
}

void selectionSort(int arr[], int n)
{
    int i, j, min_idx;

    // One by one move boundary of unsorted subarray
    for (i = 0; i < n-1; i++)
    {
        // Find the minimum element in unsorted array
        min_idx = i;
        for (j = i+1; j < n; j++)
          if (arr[j] < arr[min_idx])
            min_idx = j;

        // Swap the found minimum element with the first element
        swap(&arr[min_idx], &arr[i]);
    }
}

/* Function to print an array */
void printArray(int arr[], int size)
{
    int i;
    for (i=0; i < size; i++)
        printf("%d ", arr[i]);
    printf("\n");
}

// Driver program to test above functions
int main()
{
    int arr[] = {64, 25, 12, 22, 11};
    int n = sizeof(arr)/sizeof(arr[0]);
    selectionSort(arr, n);
    printf("Sorted array: \n");
    printArray(arr, n);
    return 0;
}
```

Output:

```
Sorted array:
11 12 22 25 64
```

Time Complexity: O(n*n) as there are two nested loops.

Auxiliary Space: O(1)

The good thing about selection sort is it never makes more than O(n) swaps and can be useful when memory write is a costly operation.

Please write comments if you find anything incorrect, or you want to share more information about the topic discussed above

Category: Searching and Sorting

tihcar • 8 months ago

What is the impact of adding a check that min_idx is same as "i" before swap? ie value in array is already at its correct location.

8 • Reply • Share ›

Aditya Goel tihcar • 5 months ago

To handle swapping when the element is already at its correct position i.e. at i,

3 • Reply • Share ›

anand • a year ago

Which sorting is better for single link list ??
If i use this selection sort then there is any problem ???

4 • Reply • Share ›

Sur anand • 9 months ago

merge sort.... http://www.geeksforgeeks.org/m...

2 • Reply • Share ›

Zsw-seu anand • 2 months ago

```
struct node
{
int data;
node* next;
};
void SeletSort(node* head)
{
```

```
{
node* p=head,*q,*k;
int tmp;
while(p!=NULL&&p->next!=NULL)
{
q=p->next;
k=p;
while(q!=NULL)
{
if(q->data < k->data)
k=q;
q=q->next;
}
tmp=p->data;
p->data=k->data;
k->data=tmp;
p=p->next;
}
}
}
```


karan anand • 9 months ago

You can use selection sort to sort a singly linked list . There won't be any problem .

guapo • a month ago

what will be the changes in this code if we pass the arguments in the swap function by value?(passing by Value)

honey • 3 months ago

tell me why int n = sizeof(arr)/sizeof(arr[0]);
we can always count the elements of array

mast monsoon • 8 months ago

is the output for the iterations are corrected....i think after 1st iteration it is
11 64 25 12 22

Jon Snow mast monsoon • 8 months ago

It is correct. Please have a look at code. The swap function swap current element

at index 'i' with minimum in remaining unsorted array. so at first iteration 64 and 11 swapped.

• Reply • Share ›

VINYAS N R • 9 months ago

@GeeksforGeeks

the output at the end of the program seems to be wrong with respest to the input taken. Please correct.

• Reply • Share ›

GeeksforGeeks Mod → VINYAS N R • 9 months ago

Thanks for pointing this out. We have updated the output.

• Reply • Share ›

Guest • 9 months ago

@GeeksforGeeks

• Reply • Share ›

Guest • 9 months ago

@GeeksforGeeks

the output written at the end of the program seems to be wrong with respest to the input taken, please correct.

• Reply • Share ›

DS+Algo=Placement • 9 months ago

@GeeksforGeeks

you wrote that selection sort never makes O(n) swaps, but in worst case there will be n swaps, isn't that O(n)

• Reply • Share ›

GeeksforGeeks Mod → DS+Algo=Placement • 9 months ago

Thanks for pointing this out. There was a typo in the statement. We have corrected it now.

• Reply • Share ›

Guest • 9 months ago

loop can be for(i=0;i

• Reply • Share ›

suraj • 10 months ago

why in worst case number of swap is O(n)?

• Reply • Share ›

DS+Algo=Placement → suraj • 9 months ago

DS+Algo=Placement ➚ suraj • 9 months ago

for every index of array, only one swap is done....hence in worst case total n swaps are needed to do

1 ▲ | ▼ • Reply • Share ›

Divyanshu Vanwani ➚ suraj • 7 months ago

In worst Case all the numbers will be arranged in decreasing order so we have swap every number , Hence O(n) number of swaps are required .

▲ | ▼ • Reply • Share ›

Gutha Raghu • 10 months ago

what if list contains duplicates,, XOR yields zeros for those fields , which is trying to swap same location values!!

▲ | ▼ • Reply • Share ›

disqus_kEyYyedb4C • a year ago

Is it stable

▲ | ▼ • Reply • Share ›

GeeksforGeeks Mod ➚ disqus_kEyYyedb4C • a year ago

The above implementation of selection sort is not stable. To make the algorithm stable, the swap() step needs to be modified to insert the minimum element rather swapping it.

1 ▲ | ▼ • Reply • Share ›

Aditya ➚ GeeksforGeeks • 5 months ago

it would be stable if we use a condition before the swap that if (mid_idx != i) , then swap()

▲ | ▼ • Reply • Share ›

RK ➚ Aditya • 3 months ago

adding that condition doesn't make any impact on the stability of the sorting algorithm.

▲ | ▼ • Reply • Share ›

Mittal ➚ GeeksforGeeks • a year ago

Sorry, couldn't understand this. Why is that so ?

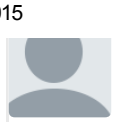
▲ | ▼ • Reply • Share ›

monty024 • a year ago

this post expalin why it is n^2 http://cs.stackexchange.com/qu...

▲ | ▼ • Reply • Share ›

kx • a year ago



why isn't the complexity = O(n!) ? since the inner loop traverses as : n, n-1, n-2, ..., 1.

• Reply • Share ›

Bhuwnesh Dhakar ➚ kx • 10 months ago

for first time when i=0 it takes = n (0 to n-1)
i=1.....=n-1 (1 to n-2)
...
...
.....i=n-2 .....=1
.....total T(n)=n+(n-1)+(n-2)+.....+1=O(n^2) (✓)
..i think you are multiplying these terms.

• Reply • Share ›

kx ➚ kx • a year ago

Wrong question..!

• Reply • Share ›

monty024 ➚ kx • a year ago

I don't think there is wrong question in the world. there are wrong answers

6 • Reply • Share ›

jagadish • a year ago

We Can optimize the Selection Sort algorithm by selecting minimum element using Heap, which takes O(logn) to get the minimum element.

The Total Time Complexity will be reduced from O(n^2) to O(nlogn).

• Reply • Share ›

Thrinadh ➚ jagadish • a year ago

@jagadish Heap takes O(n) to get a minimum element. So by using heap we again get O(n^2)

1 • Reply • Share ›

Legolas ➚ jagadish • a year ago

If we can get time complexity of O(n logn) then we should use selection sort instead of merge sort.!

• Reply • Share ›

Guest • a year ago

what if we want to order numbers in array in increasing order through selection sort?

• Reply • Share ›

jagadish ➚ Guest • a year ago

Pick the Maximum element from the unsorted array each time and place it in the front of sorted array.

• Reply • Share ›

Jai • a year ago

ama12, Cycle Sort is a better algorithm in terms of memory writes. selection sort is good for nothhing

• Reply • Share ›

jagadish Jai • a year ago

Selection sort is good if we use with the combination with heap, which results time complexity O(nlogn).

• Reply • Share ›

ama12 • a year ago

Is selection Sort the best algorithm in terms of memory writes?

• Reply • Share ›

ama12 • a year ago

Is selection sort the best algorithm in terms of number of swaps?

• Reply • Share ›

Sumit Khatri ama12 • a year ago

yes, it is the only sorting technique which requires O(n) swaps even in worst case

• Reply • Share ›

jagadish ama12 • a year ago

Quicksort even better then selection sort in terms of number of swaps.

• Reply • Share ›

Ravi Teja Kaveti jagadish • 9 months ago

Quick sort will take more number of swaps than selection sort in worst case conditions. We usually consider worst case conditions so selection is better than Quick in number of Memory writes. correct me if i am wrong .

1 • Reply • Share ›

Sumit Khatri jagadish • a year ago

no, quick sort requires more swaps than selection sort in worst case

• Reply • Share ›

ananymous Sumit Khatri • 4 months ago

no more swaps quick uses recursive so it takes less time

• Reply • Share ›