

Forum:[Hack Forums](#)

Thread: [Need help about Selection Sort in Java !](#)

Teleforce - October 13, 2017, 1:17 am

Hi guys,

I need answer for this question if your guys know the answer.

- What is the big-O running time for selection sort if all of the elements are equal ?

I appreciate your help ?

GammaG - October 13, 2017, 1:37 am

Can you post a code snippet of the algorithm you use?

I'd say depends on how much loops are in there and how the comparison of the elements is implemented.

Teleforce - October 13, 2017, 3:05 am

(Oct 13, 2017 01:37 AM)GammaG Wrote: [\[CITE\]](#) Can you post a code snippet of the algorithm you use?

I'd say depends on how much loops are in there and how the comparison of the elements is implemented.

For example this simple program

Code package Selection_sort;

```
public class Selection_sort_john{
```

```
    public static int[] Selection(int[] New_array){
```

```
        for (int i = 0; i < New_array.length - 1; i++)
```

```
        {
```

```
            int index = i;
```

```
            for (int j = i + 1; j < New_array.length; j++)
```

```
                if (New_array[j] < New_array[index])
```

```
                    index = j;
```

```
            System.out.println(index);
```

```
            int smallerNumber = New_array[index];
```

```
            New_array[index] = New_array[i];
```

```
            New_array[i] = smallerNumber;
```

```
        }
```

```
        return New_array;
```

```
    }
```

```
    public static void main(String a[]){
```

```
        int[] array_1= {5,5,5,5,5,5,5,5};
```

```
        int[] array_2 = Selection(array_1);
```

```
        for(int i:array_2){
```

```

        System.out.print(i);
        System.out.print(" ");
    }
}
}

```

Solixious - October 13, 2017, 5:29 am

The run time of selection sort is typically $O(n^2)$, where n is the size of the input.

Teleforce - October 13, 2017, 7:21 am

(Oct 13, 2017 05:29 AM)Solixious Wrote: [\[CITE\]](#) The run time of selection sort is typically $O(n^2)$, where n is the size of the input.
could you give me more details ?

?

UB3R5W4G0V3RL0RD - October 13, 2017, 9:44 am

(Oct 13, 2017 07:21 AM)Teleforce Wrote: [\[CITE\]](#) could you give me more details ?

?

I'll just copy an explanation from another website which I found fairly comprehensible. The site itself explains the most common ones and how to find them, along with some nice gifs to help visualize them.

Link: corejavainterviewquestions.com/idiots-guide-big-o/
(Can't post links yet, sorry! Just add www to the beginning)

tl;dr Wrote: $O(n^2)$ /Quadratic Complexity: Things are getting extra slow. 1 item takes 1 second, 10 items takes 100, 100 items takes 10000.

Explanation Wrote:The Bubble Sort algorithm is everyone's first algorithm in school, and interestingly it is quadratic complexity. If you need a reminder; we go through the list and compare each element with the one next to it, swapping the elements if they are out of order. At the end of the first iteration, we then start again from the beginning, with the caveat that we now know the last element is correct.

Imagine writing the code for this; it's two loops of n iterations.

PHP Code

```

<?php
public int[] sort(int[] toSort){
    for (int i = 0; i < toSort.length -1; i++) {
        boolean swapped = false;
        for (int j = 0; j < toSort.length - 1 - i; j++) {
            if(toSort[j] > toSort[j+1]){
                swapped = true;
                int swap = toSort[j+1];
                toSort[j + 1] = toSort[j];
                toSort[j] = swap;
            }
        }
    }
    if(!swapped){

```

```
break;  
}  
} return toSort; }
```

This is also a good example of best vs worst case. If the list to sort is already sorted, then it will only take n to sort; a single iteration will do. However, in the worst case where we have to go through the list n times and each time looping another $n - 1$ (first loop) which is slow.

You may notice that it's technically less than n^2 as the second loop decreases each time. This gets ignored because as the size of the data set increases this impact of this becomes more and more marginal and tends towards quadratic.

Solixious - October 13, 2017, 12:02 pm

(Oct 13, 2017 07:21 AM)Teleforce Wrote: [CITE](#) could you give me more details ?

?

Certainly. The outer loop in the selection sort runs n times. For each time the outer loop is executed, the inner loop is executed up to a maximum of n times and average of $n/2$ times. Thus, the total number of times the statements within the inner loop is executed is averagely

Code $n * n / 2$

or

$(n^2) / 2$

That is in order of n^2 , hence the run time complexity of selection sort is $O(n^2)$.

You may google for theoretical definition of run time complexity and Big O notation to understand what it means exactly if you are having trouble with that.

Regards.