

The background of the image is a dark, textured surface with a grid of numbers in a light gray font. The numbers are arranged in a repeating pattern, with some numbers being positive and some negative. A prominent yellow wavy border frames the central text area.

Algorithms in Computer Science

CHAPTER 9

Sorting Algorithms

Introduction

Imagine you had to search through a list of names to find a specific one, would you rather that list be in order or not in order? Obviously, you would want an ordered list. This being the case, computer scientists came up with multiple ways, some more efficient than others, to sort the large amounts of data they had to manage. They invented sorting algorithms that put elements of a list in order. There are multiple sorting algorithms out there, but this book will focus on four (4) of them: insertion sort, selection sort, bubble sort, and merge sort.

Insertion Sort

The insertion sort algorithm is a simple algorithm that sorts the elements in a list one at a time. It assumes the first element is already sorted and starts with the second one. If the element being sorted is less than the one before it, it is moved over to the left so that the two are in order. The algorithm then checks again if the element being sorted is less than the one before it. If that is the case, it is moved again to the left. This keeps going until the element being sorted is the first element or is bigger than the one before it.

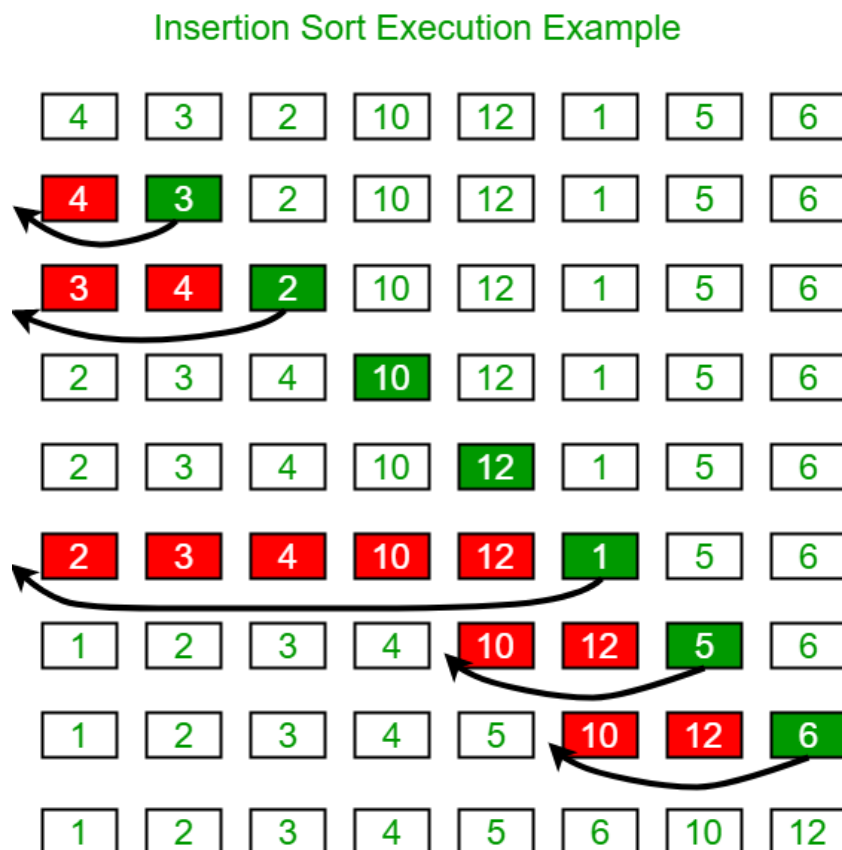


Figure 1

Here is the code for an insertion sort in java.

```
public class InsertionSortExample {  
    public static void insertionSort(int array[]) {  
        int n = array.length;  
        for (int j = 1; j < n; j++) {  
            int key = array[j];  
            int i = j-1;  
            while ( (i > -1) && ( array [i] > key ) ) {  
                array [i+1] = array [i];  
                i--;  
            }  
            array[i+1] = key;  
        }  
    }  
}
```

Selection Sort

The second sorting algorithm we will cover in this book is the selection sort algorithm. A selection sort is an algorithm that, in order to sort a list, creates a new list and copies the elements of the old list and adds them in order to the new list. This sorting algorithm searches the smallest or minimum element in the list that is being sorted and inserts it into a new list. It then goes back to the old list and

Here is the code for a selection sort in java.

```
public class SelectionSortExample {  
    public static void selectionSort(int[] arr){  
        for (int i = 0; i < arr.length - 1; i++){  
            {  
                int index = i;  
                for (int j = i + 1; j < arr.length; j++){  
                    if (arr[j] < arr[index]){  
                        index = j; //searching for lowest index  
                    }  
                }  
                int smallerNumber = arr[index];  
                arr[index] = arr[i];  
                arr[i] = smallerNumber;  
            }  
        }  
    }  
}
```

Bubble Sort

The bubble sort algorithm takes a simpler approach to the problem. This algorithm compares every two elements that are next to each other in a list and swaps them if they are not in order. It compares the first and second elements, swaps them if need be, then the second and third and so on. Once it is done going through the whole list, it starts over from the beginning and repeats the process. The algorithm keeps doing this until the whole list is sorted. But how does it know when to stop, you

might ask. The algorithm knows to stop when it has gone through the whole list again and there were no swaps to be done.

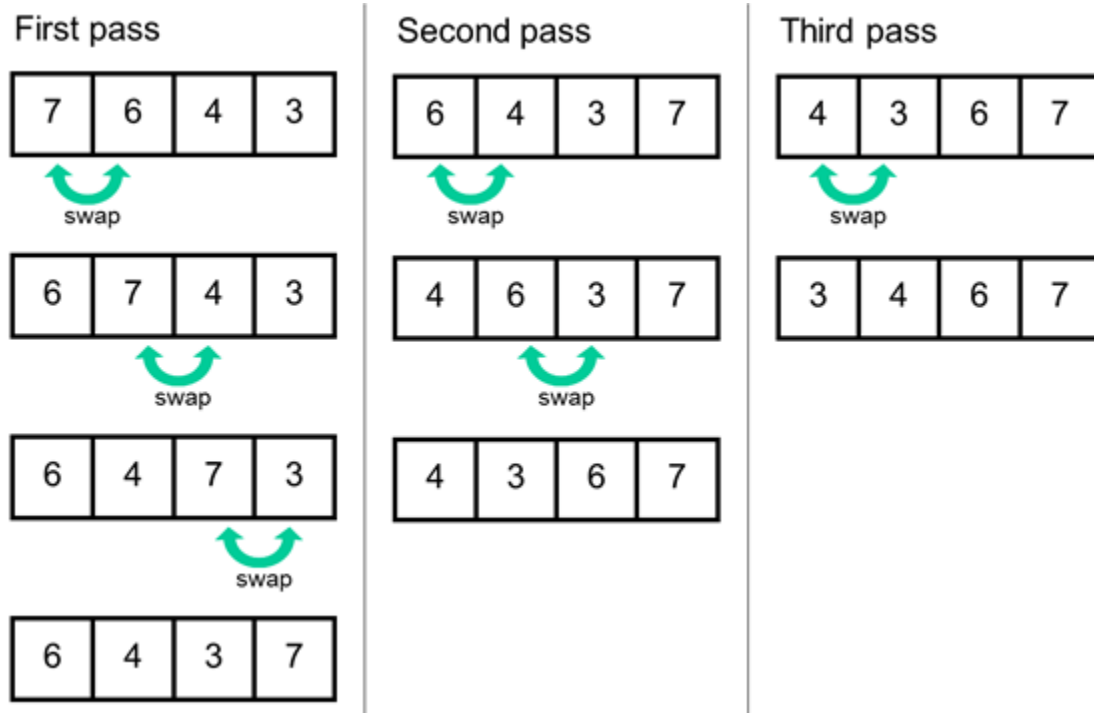


Figure 3

Here is the code for a bubble sort in java.

```
public class BubbleSortExample {  
    static void bubbleSort(int[] arr) {  
        int n = arr.length;  
        int temp = 0;  
        for(int i=0; i < n; i++){  
            for(int j=1; j < (n-i); j++){  
                if(arr[j-1] > arr[j]){  
                    //swap elements  
                    temp = arr[j-1];  
                    arr[j-1] = arr[j];  
                    arr[j] = temp;  
                }  
            }  
        }  
    }  
}
```

Merge Sort

The last sorting algorithm we will look at is the merge sort algorithm. Instead of sorting the whole list, the merge sort algorithm takes the divide and conquer approach and divides the list into smaller chunks. The list is first divided in half and then, each half is halved again until each chunk has only one or two elements. If a chunk has two elements, they are swapped if they are not in order. The

algorithm then merges all the chunks back together in order, two chunks at a time.

Hence the name: merge sort.

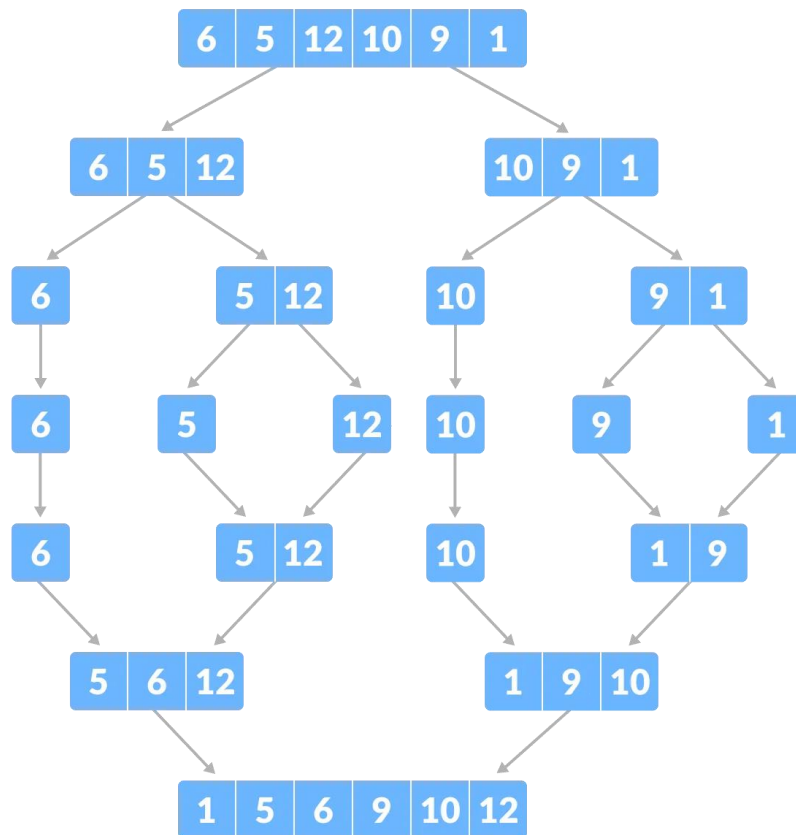


Figure 4

Here is the code for a merge sort in java.

```
void merge(int a[], int beg, int mid, int end)
{
    int i, j, k;
    int n1 = mid - beg + 1;
    int n2 = end - mid;

    int LeftArray[n1], RightArray[n2]; //temporary arrays

    /* copy data to temp arrays */
    for (int i = 0; i < n1; i++)
        LeftArray[i] = a[beg + i];
    for (int j = 0; j < n2; j++)
        RightArray[j] = a[mid + 1 + j];

    i = 0; /* initial index of first sub-array */
    j = 0; /* initial index of second sub-array */
    k = beg; /* initial index of merged sub-array */

    while (i < n1 && j < n2)
    {
        if(LeftArray[i] <= RightArray[j])
        {
            a[k] = LeftArray[i];
            i++;
        }
        else
        {
            a[k] = RightArray[j];
            j++;
        }
        k++;
    }
    while (i < n1)
    {
        a[k] = LeftArray[i];
        i++;
        k++;
    }

    while (j < n2)
    {
        a[k] = RightArray[j];
        j++;
        k++;
    }
}
```

Conclusion

Programmers have to use many tools and algorithms in order to keep data organized. Sorting a list might seem like a simple thing but, when it comes to coding, there are multiple ways to get it done. Software engineers have to think about how long it takes a computer to finish a task. So, they came up and are still coming up with multiple different ways to do things. Sorting algorithms are an example of this. Each one is different, and some are more efficient than others.

Practice Questions

1) On which element in a list does an insertion sort start?

2) Why does the selection sort need a search algorithm?

3) True or False:

The bubble sort algorithm goes through a list only once.

a. True

b. False

4) What is being merged in a merge sort?

Solutions

- 1) An insertion sort starts with the second element in the list.
- 2) The selection sort needs a search algorithm to find the smallest element in the list every time it goes through it.
- 3) False.
- 4) In a merge sort, the divided chunks are merged back together in order.

References

Easton, T. (2017) *Numbers*

<https://unsplash.com/photos/faixctm2YRQ>

Geeksforgeeks.com *Insertion Sort*

<https://www.geeksforgeeks.org/insertion-sort/>

hackerearth.com *Selection Sort*

<https://www.hackerearth.com/practice/algorithms/sorting/selection-sort/tutorial/>

Stanley, A. (2021) *Bubble Sort*

<https://medium.com/austins-software-engineering-journey/bubble-sort-b2f0d63e38f7>

Programiz.com *Merge Sort Algorithm*

<https://www.programiz.com/dsa/merge-sort>

javatpoint.com *Insertion Sort in Java*

<https://www.javatpoint.com/insertion-sort-in-java>

javatpoint.com *Selection Sort in Java*

<https://www.javatpoint.com/selection-sort-in-java>

javatpoint.com *Bubble Sort in Java*

<https://www.javatpoint.com/bubble-sort-in-java>

javatpoint.com *Merge Sort Algorithm*

<https://www.javatpoint.com/merge-sort>