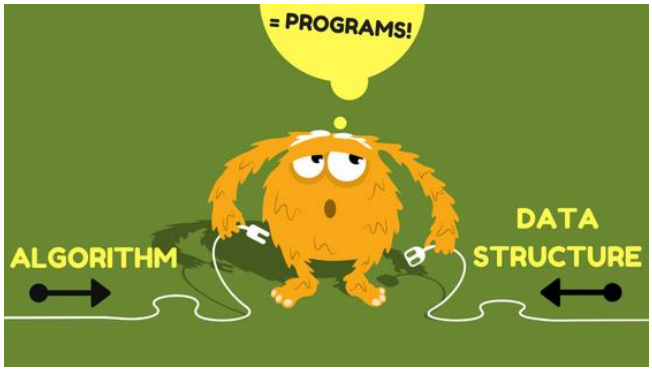


DATA STRUCTURE & ALGORITHM

DSA

ABSTRACT

To know about the how algorithm work with data structure

Mr. Thihan Hein

DSA

Contents

Data Structure and Algorithm

Data Structure

Data structure is a kind of data type that announce the type of the variable such as integer, float, array and so on. Data structure can store the data. For example you can add data with an array. You can add data whatever you want. But how will you select those data. You can select the data. But the data will be like a trash and for a normal user can be confuse. Data must be need clear for the end user. For example, there are many search engines. Among them, Google search engine is most useful. So why Google engine is most useful? Because of the algorithm, it's more powerful. Algorithm is to learn about the program to make more performance as a backend coding.

Algorithm

Algorithm is a detailed step by step method for solving a problems. It is well developed and good approach to solving complex problems. For example, make sorting the data. Without algorithm you can also sort data. But it's difference. It might be take more time to sort. So algorithm is to make a good performance program. As an example with Google Search engine. Why it is used by all people? Because of the search algorithm, it can make more performance of the search result. So Algorithm is to find the best way solution of the program performance. In easy way, every program has their own original design, color and structure. Like fast sorting, fast searching, fast saving and so on. So algorithm is one of the design, color structure of the program.

So for example with the video call application like Viber, Discord, Messenger and so on. Before inventing those application, we connect with the satellite connection to connect each other. In satellite, we have to wait a minute to hear the voice. When hearing voice, we can't hear clearly the voice. It has some barriers noises. Because it through the satellite and at that time developer don't know the better algorithm. So after that, developer think about the algorithm to clear the barriers noises. Viber, Discord, Messenger and such like application has different algorithm to prevent from the external voice barriers. So Algorithm is to make the application better with the better performance. That is why we learn about the algorithm.

Sorting Algorithm

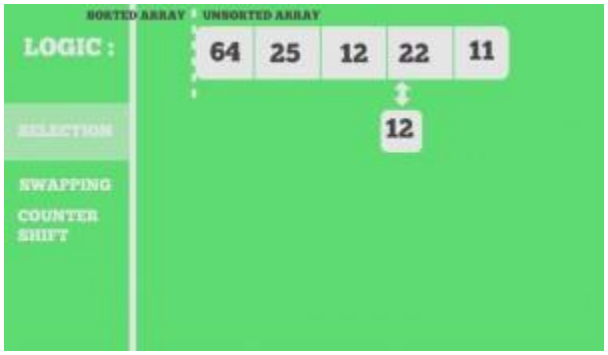
There are many type of algorithm such as sound clear algorithm, searching algorithm and so on. Among them I want to explain about the sorting algorithm. Sorting Algorithm is a kind of algorithm that find to make sort fast. You can make sort without algorithm. But sorting can make you to make better performance in sorting. There are many type of sorting algorithm and I want to explain their benefits and limitation.

Selecting Sort

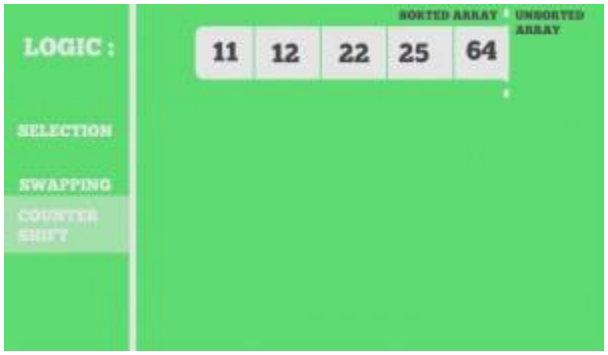
Selecting sort is to select a small number and put at the beginning. Selecting sort is easy for a junior programmer and easy to understand. For example there has an array room and all the room are unsorted. With selecting sort, it will select smallest number that has in that array and put it at the first index of the array and the first array will reach to the smallest number room. After that, the first index of the array will lock and it will start from the next room of the lock array index.

So it has the array 64,25,12,22,11. Now let's sort with for low to high by using selecting. First select 64 and check with beside of the array room.

Now the select number is 25. And check the beside of 25 array room.



Now the select is 12. And check the beside number of array room. If the select number is less than beside of the array room. It doesn't change and keep checking. Now the select number is 11. And it will swap 11 and 64 and the swap room was locked and continue selecting.



As a real world example, it will use in student grade result in each classes. Assume that each class has 60 or 70 students. Sorting them according to ID is not an easy task. Here, we can simply use selection sort to sort these 60 or 70 students in no time. By swapping the IDs that came in front with respective to those who are not in order. Although it may seem confusing, it can make sorting easier for those who are supposed to sort this.

Benefits of Selecting Algorithm	Limitation of Selecting Algorithm
• **The Sorting is very fast**	• it will doesn't effect in so many data

• **It use less memory**	• If the data is too much it can take much time
• **It doesn't need temporary storage to store an index**	• It doesn't match with a huge business

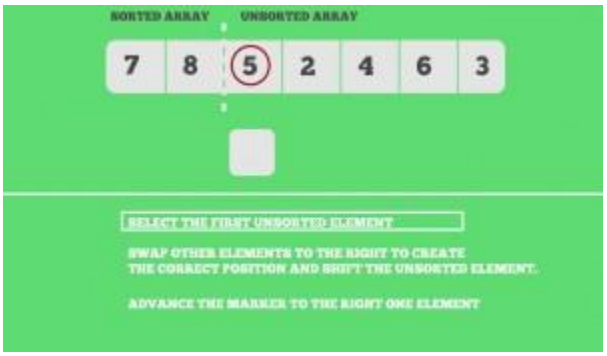
Selecting sort is good for only if the data is less than 60 or 70. It can't handle 100 of data. Because it can take much time to finish it in a short time. And it doesn't support in a large business. In my example, it is suitable for a classroom grade of student. Because, it only have 60 or 70 students. In short, selecting sort is good for less data, if not it will take long time to make sort.

Inserting Sort

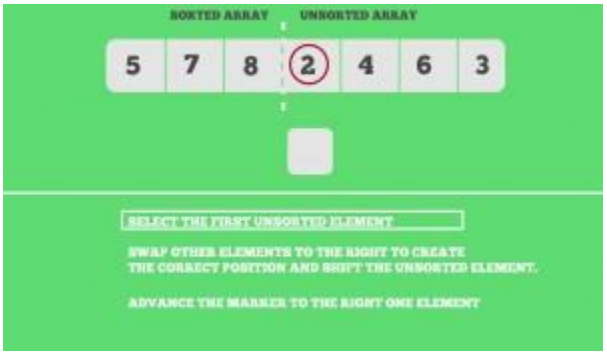
Inserting Sort is a kind of algorithm that select an array index number and check with the right array indexes. And it will catch as a temporary memory after check with right index. If the right index is correct it will stop mark temporary and put a real number. If not it continue store temporary and check with the right index rooms.

So here is the number 7,8,5,2,4,6,3 array. So first it select one number and check with the front arrays. So 8. Check with the 7. If true, it doesn't make any changes.

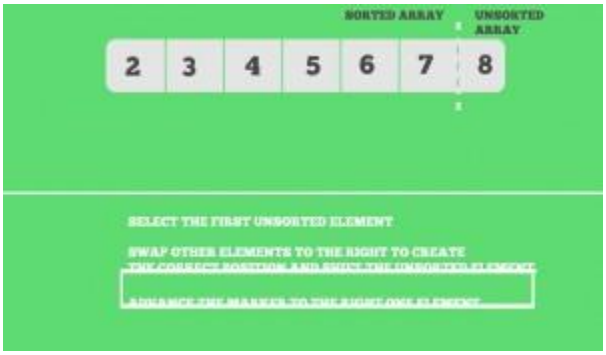
And now it's 5. Check with the left side of the array room. Check with 8. False and 8 was place in 5 array room. But in 8 array room 5 was stay as a temp memory to check with the rest array room. So 5 check with 7 false so swap.



After that array count is plus one and another number will be selected as 2. It will check with the rest of the array room and stored as temporary memory until it's true or nothing to check with other. The rest array room will check like this.



Finally, the array is sorted. As conclusion. Insertion sort is to select the number and check the front of the array room until it's true. And if the array has need to check it will store temp to check with the rest.



With a real world example, a card games, poker as an Microsoft games. It will give you unsorted number and you need to make with minimum number to maximum number. It use Inserting sort to make an order. Inserting sort is also simple and it can easy to understand.

Benefits of Inserting Sort	Limitation of Inserting Sort
• **It is simple to understand and easy to use**	• It doesn't perform as well as other sorting. Because it repeated again even the condition is true
• **It comfortable with small business**	• Doesn't support for large business
----------	• It store temporary data, so if a data is much it can damage and harm to hardware

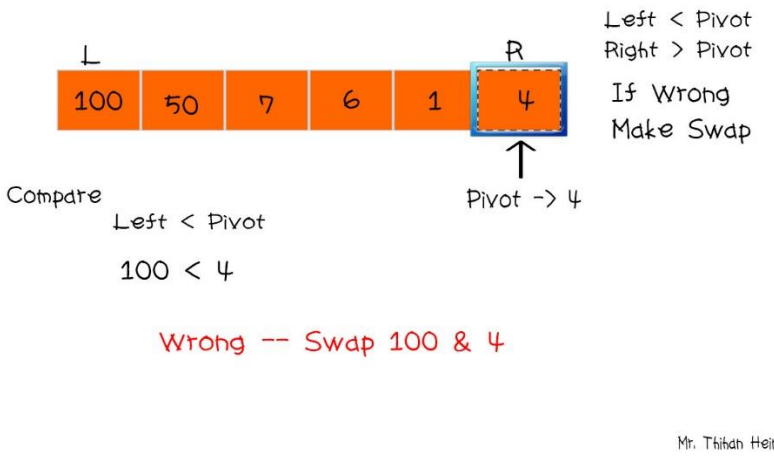
Inserting sort is also good for the small business and it is only good what to do exactly. For example, sorting poker (Cards) program game. So we know how many poker does it have and the selecting poker will little to play. So we can know the counter exactly. At that time, we can use inserting sort to sort the poker program. And it doesn't support large amount of data and doesn't know the total counter.

Quick Sort

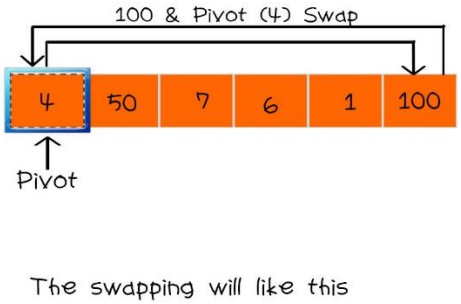
Quick sort is a divide and Conquer algorithm. It pick one of the number that has in the array index as pivot. After getting pivot it make partition into left and right. There may be different for quick sort for getting pivot. The pivot can get by the following statement. Be careful, if you choose one of those steps, the other steps should also same with the previous step.

1. The first element will always be set as pivot.
2. The last element will always be set as pivot.
3. A random element can be picked as pivot.
4. Median can be picked as pivot.

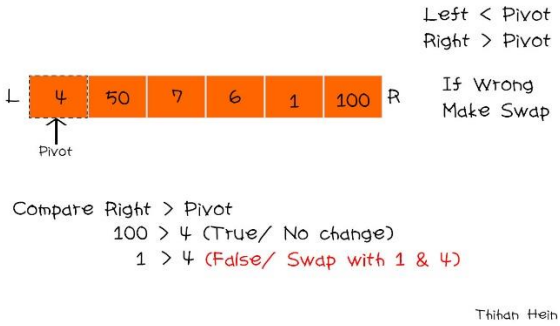
In this report, I want to choose the pivot as a last element. Here is an example with some number.



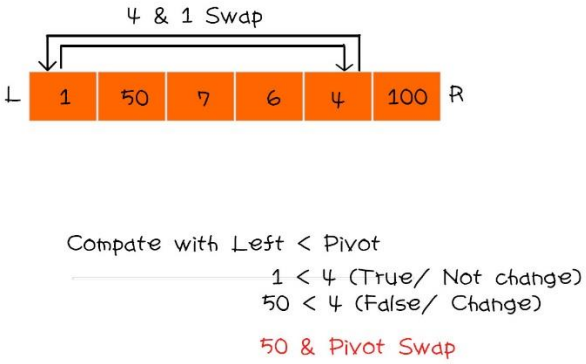
So I will take 4 as a pivot. In this statement it will have a condition. If the number of the array index is less than pivot, it will swap. If not it doesn't change. So pivot is 4. And it will check with the first number of array index (100). 100 < pivot that is true. So both room will swap. And it will change as follow.



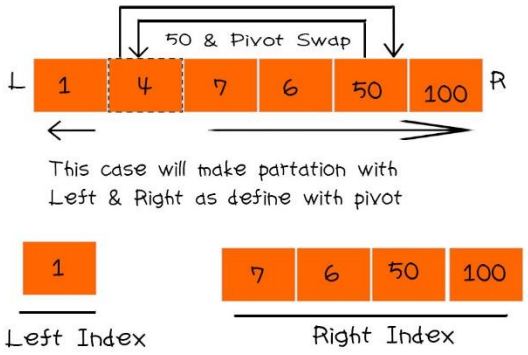
After swapping pivot and an array index room, it will check with Left and Right as define with pivot. Now 4 is pivot. 100 > pivot (4) is true, so it doesn't change. 1 > pivot (4) is not true so, it will make swap with 1 and 4. So it will be as shown in figure.



So swap like that. Now it's also make partition again as Left and Right with same pivot (4). it will check both left and right. 1< pivot (4) is true so, it doesn't change. And 50 < pivot (4), it doesn't true. So 50 and pivot will swap. Now the sort will like this.



If the pivot is in the middle it should divided into two partition that has on the side with pivot. So the partition will be like this

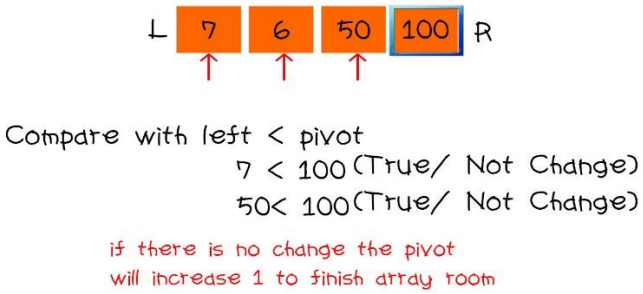


So the right side of the room will do with the last pivot and it will do same steps with uppercase.

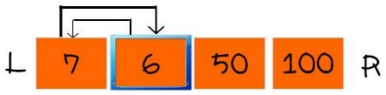
7 < pivot (100) is true. So it doesn't change. And the rest will also same it. if not found it the array will reduce from the right side. So at this time 50 will pivot.

7 < pivot (50) is true so doesn't change anything and keep going. And 6 will take as pivot.

7 < pivot (6) is false, so swap those number as following



Now 6 is pivot and check again with right side. 100 > pivot (6) is true so doesn't change nothing and the rest will same with. After that, the left index and right index will combine together as following.



Compare with left < pivot
6 < 7 (False/ Change)

And it will also take pivot for last index (100). 1 < pivot (100) is true, so doesn't change nothing and the rest will do like this and finally the array room will finally sorted.

Quick sort is an very effective sorting algorithm and mostly used in nowadays. Quick sort is to sort with pivot with left and right with adjustment for the sorting. Quick sort fast enough and good performance thank selection and inserting sort.

As a real world example it use in the sorting that want to know the result of the order number by descending or ascending. It is mostly use in supermarket for sorting the buying or order process. Because of the data of the order, it can have more than 1000, quick sort can handle it and for the customer who doesn't have patient is the best way with the quick sort.

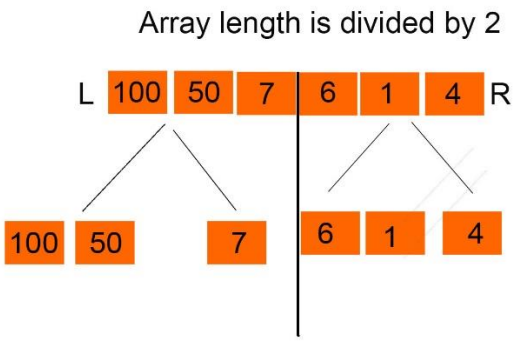
Benefits of Quick Sort	Limitation of Quick Sort
• **It is very effective in sorting and it is use widely**	• It need more space to finish the sorting
• **Effective on both small and huge business**	• It need much knowledge of Sorting
• **If a developer understand and can implement quick sort, he is really and certain to apply a job**	• It use much memory because it need many steps to finish
--------	• Some of the parts theory are difficult to understand

Merge Sort

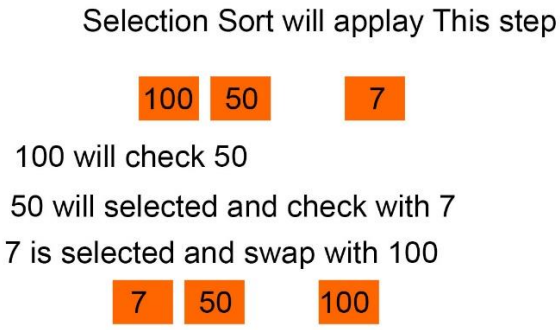
Merge sort is a kind of sorting algorithm that is as efficient as quick. Merge sort is combine with Selecting that divide and conquer algorithm. It divide array room into two half until it get the last one. It is easy to understand for the merge sort. Here is an example of merge sort. The array is divided by 2 and separate the result. If the result is odd, it take as an even. And it will separate until the room is last 1 in separation. Here is an example.

100 50 7 6 1 4

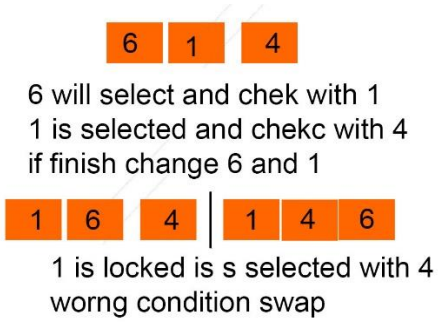
This array room is divided by 2 and it will separate as shown in figure



It doesn't finish because the all the room is doesn't last as a one. So it need to do same with first steps that divide by 2.

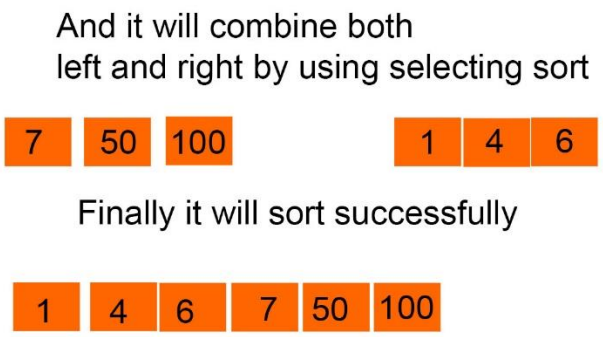


And it's still need to finish so it also divide by 2.



So all the array room are separated and there has no room to separate. So in this state, selection sort will do this process for each partition. For the left index, it will make selection sort for both left and right.

After making selecting sort, both left and right, it will make sort for both left and right,



Now the array is sorted with faster.it is easy, isn't it? Merge sort is combine with divide and selecting sort to complete Merge sort.

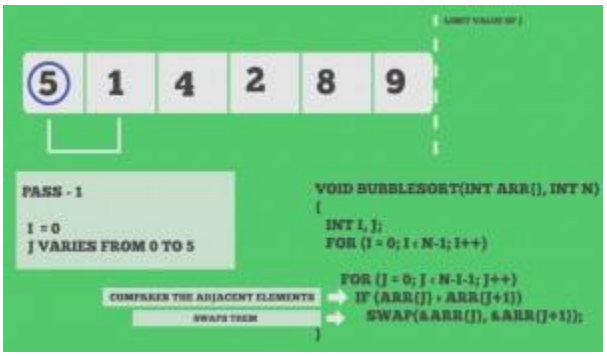
As a real world example, merge sort is mostly use if the data is large and the sorting time is need to finish hurry. For example, A University Student Exam Result, It should use Merge sort. Because a university can have many student and it will have many grade. At that time, selecting sort is not suitable and quick sort is also. Even quick sort can take long time to finish. So merge sort is suitable for this process because it process is simple and very effective because combining divided and selecting sort. Here is benefits and Limitation of Merge Sort.

Benefits of Merge Sort	Limitation of Merge Sort
• **Main benefits is that it can apply for huge data and can get fast processing**	• It is less efficient than Quick sort
• **Processing is very fast and comfortable with both huge business and small business**	• It can have more memory usage
• **It is easy to understand than quick sort**	-----

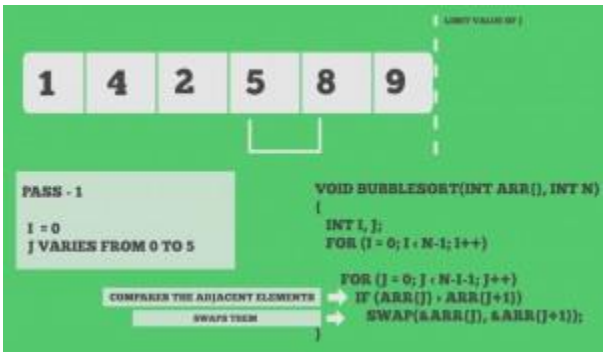
Bubble Sort

Bubble sort is the simplest sorting algorithm by swapping each other if they are not sorted. It can refereed to sinking sort which repeatedly steps through array. It can repeated until the list is sorted. It swap side of the room and it can repeat process until its finish. So here is a clear example. We have 5,1,4,2,8,9. And we will sort it by using bubble sort. Remember bubble sort is check the both side of each array room.

So first two array room will be 5 and 1. 5 will selected and check with the right side of array room. If the condition is false it will swap. So now it will make swap with 5 and 1.

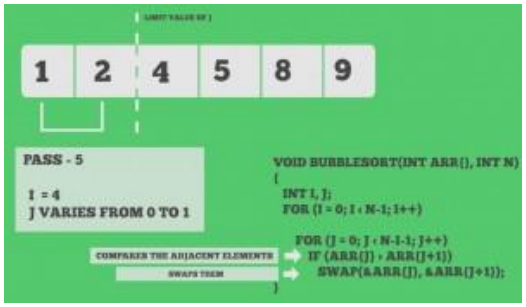


And now it again check the swapped number and right side of the room. It will be 5 and 4. 5 is swapped number with 1. So 5 and the right side of the 5 is 4. So it will check. False! Swap. So 5 and 4 ware swapped. And it will have as shown in figure.



Now the swapped number (5) and right side of the swapped array room (8) will checked. If the condition is true it doesn't change and the room will be next selected room. The selected room 8 and right side of the room were checked. If true it doesn't change nothing and the array count will reduce. And it will repeat process form start.

So it will start with 1 and right side of the array room 4. If true doesn't change and index will increase. So it will be 4 and right side of array number is 2. The condition is false so it swapped. So the swapped number (4) and right side of the array room 5 will check. The condition is true so doesn't change nothing and index will increase. So 5 and 8 will check. Condition is true doesn't change nothing and array count will reduce and lock. And it will start again until array count doesn't finish. It seems it will sorted. But it doesn't finish. It will continue the size of the array even all the room are sorted. Finally you will get sorted number.



As a real world example it is suitable use in school report number or office report number sorting in class. In class it only have 60 student each class in school. Because the number can be low or big. We

can't know exactly. Because of repeating process of the bubble sort it is suitable. If the big report number is insert in the middle, it can easily sort because it work until the array size is finish.

Benefits of Bubble Sort	Limitation of Bubble sort
• **First is easy to understand**	• It can use between the data is less than 300 because of the repetition process.
• **Best performance than inserting and selection sort**	• Sorting time is increase if the data is too much or the process is repeat again and again
• **It doesn't need temp memory to store so it doesn't need much garbage**	• It can have less efficient if the data is input again.

Comparison for Sorting Algorithms

No	Selecting Sort	Inserting Sort	Bubble Sort	Quick Sort	Merge Sort
•	It is very fast sort if the data are less than 10.	It can be fast if the data is less than 25	It might be fast when data is as same as inserting sort	It can take much time to sort but it is very effective for huge business	Can take much time in dividing array to sort if the data are too large
•	Easy but useless for huge business	It repeated until the sort is not finish so it can take long time		It can take pivot as different condition. So it can different some condition by getting different pivot	Combine with selecting sort. So after dividing, it do selecting sort.
•	Select the smallest number of the index and swap and lock	Take a number and check with the all left index rooms	Check the two index that is beside each other and swap	Take pivot and divide left and right to sort and combine and sort again	Separating all array room until it reach to one and make selecting sort.
•	Thinking flowchart is easy	Complex than selecting sort because it has temporary memory	Same as Inserting sort	Advance level of Programmer	Easy to understand to design merge sort
•	If the data are too much, it can have memory allocated	Memory allocation is also appear in inserting sort if the data is too much. Because it need temp	It can be less memory allocated because it move room by one.	It can be less memory allocated because it check with pivot with the rest room	It can have a little bit allocated because it need to divide and make temp and do

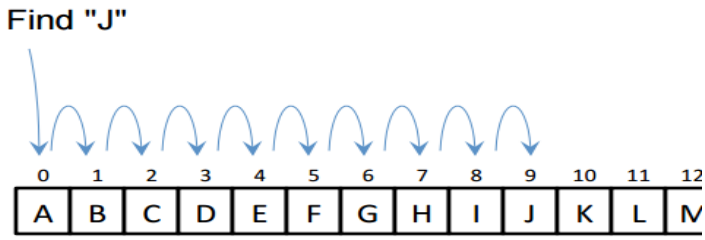
		file to check with other			the selecting sort. Sometime it can have memory allocated but sometime it doesn't make memory allocated.

Search Algorithm

Search algorithm is step by step which used to locate the exact data from the collection of data. it is a kind of computing. Search algorithm help to make faster searching. For example, there are many search engine. Among them Google is very famous. Because Google search algorithm is the best. That why searching algorithm is important. There are many type so search algorithm as following here.

Linear Search

Linear search is a kind of searching algorithm which check with each array room until it reach correct keyword. It check each element of the list for the target value until a match is found or until all the element have been searched. For example, there are a number array room which has 1 to 10. If you enter 6 that you want to find. It will search all the array room. It will check from start of the array room until correct. And it will return array room number if found. Here is clear example for linear search.



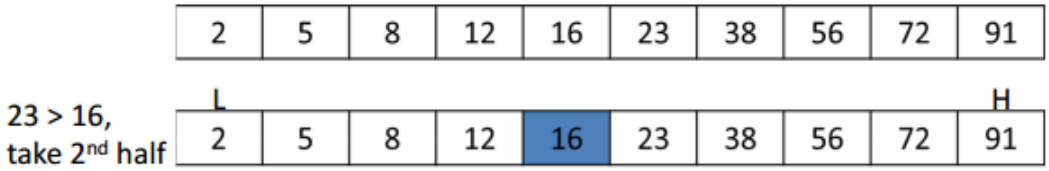
Benefits of Linear search	Limitation of Linear Search
• **If the data is less, searching is very fast.**	• If the date is large more than 100 it can take time to search. Because it check the room by one.
• **It doesn't need any temp file to memorize because it check the room by one. If it doesn't match it skip.**	• If the data are same like '3' in is array room of '4' and also '3' is in array room of '6' if the data found the first array room '4' is taken and it doesn't check the other room. So the data can't correct if the data are same.
• **It is easy to understand for the programmer.**	•
• **It doesn't need sorting to search.**	•

As a real world example, in a library to search old books or ancient books. It is very effective way to use linear search. Ancient books are not more than 100. So it can use linear search and it doesn't need sorting and the ID of the books can't same. So it can use in library to find ancient books. Not modern book.

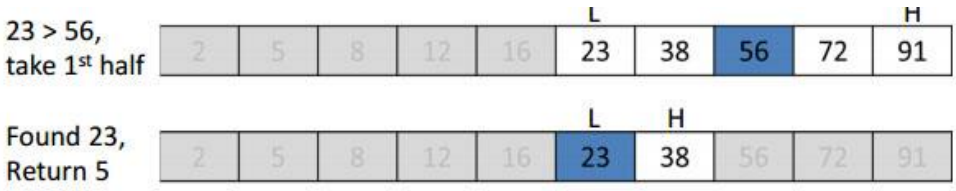
Binary Search

Binary search is a kind of searching algorithm to search an item from a sorted list. The method of binary search is dividing for each portion of a list. The main point of binary search is it have to be sorted. The array must be sorted. If not it is difficult and it rule is only for the sorted. Binary need sorted because it check the number with greater or less than the find number. Here is a clear description.

It take the search number is greater than the middle and it's true, it take right. If not take left. Here is 10 array room and you want to find 23 in that array. So first make tow partition with same half. And with the half number seem a middle number. Check is 23 is bigger than 16? Yes, so take right side of the array room.



And make same partition with the first stage. So take middle number 56. Does 23 is greater than 56? No. so take the left side of the array.



So it will have two array room. So also divide same with the first stage. And check. If the number doesn't match check with the other room. If the data is match bingo. It will return the array number room. That is why this searching i=need sorting because it check divide and conquer like merge sort. If the data doesn't match it won't show nothing. But main point it must be sorted. If not you can't find matches.

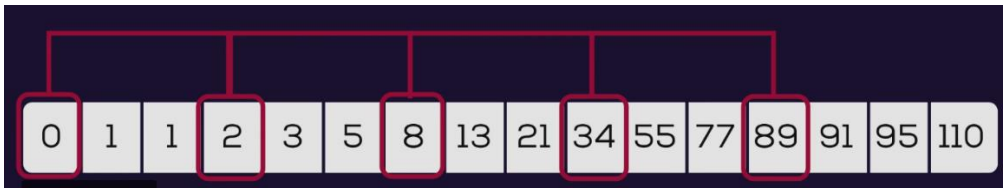
Benefits of Binary Search	Limitation of Binary Search
• **It is faster than linear search and it can use in the middle range of searching**	• The array need sorted if not it can't use
• **It is fairly simple algorithm because easy to understand**	• If the data is too much more than 1000 it can take time to finish.
• **It doesn't have memory garbage because it check only greater or less**	• It is more busy than the other searching because if the data is not sorted you have to sorted first and you have to decide the sorting algorithm to use binary search. So it can also take time to implement.
•	•

As a real world example, a good example is online ranking game. To find top players. They allow only 100 for top players. And a sponsor want to find not too high and not low rank. So if the input the score number. It will search by divide and conquer for the ranking point. If it's match, it will show.

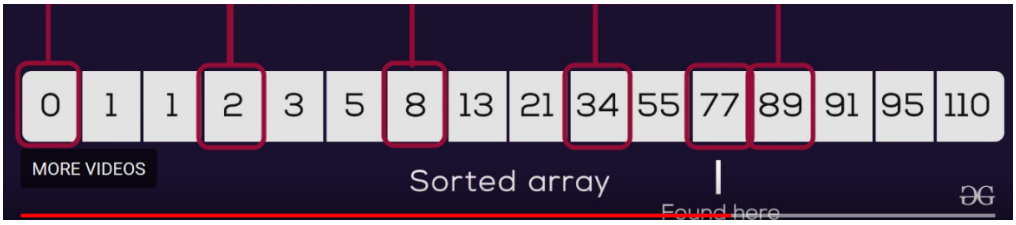
Jump Search

Jump search is also known as block search which represent to a search algorithm for a list. So it jump fro the block and find the matches by checking grather or less than the search number. It combine with linear search. It check with the square root of the array size and the result size is that it will jump. If the result size is 3 it will jump for three block for each size and check the jump room with search key. One main point is, the array must have been sorted. Because it skip other array room. Here is clear example.

Here is 16 array size. The square root of array size is 4. That number will jump number for each block. And you want to find 77. And this array size is sorted. So first it will jump four block. So 77 search number is greater than the jump number '2'. If true jump next four block. It will go until false.



Now it reach to the jump block 89. Search key 77 doesn't greater than 89. False. So it will apply linear search. It will reduce the back until it correct. So previous size of 89 is 77. If they are match bingo. You will find it successfully.



Benefits of Jump Search	Limitation of Jump Search
• **Doesn't need any cache to find.**	• Array need be sorted because of it's rules
• **Sorting is very effective and fast because it jump to other block.**	•
• **Easy to understand for newbie programmer**	• If the data are the same it only show one time
• **It can use in large process like finding over 1000 of data.**	•

As a real world example, it can apply in finding staff list. Staff can only have less than 50 in a company. This can have very effective for finding in staff with id. Most staff search program of algorithm will use jump search. Because data is less and very effective than linear and binary search.

Sorting and Search for Assignment

Sort and Search Algorithm Name	Description
Merge Sort	For the book data it will use merge. Books data must find quickly and books can have too many data in the text file. And merge sort can also handle the data. And it can have performance for sorting. We already know that selecting sort is fast in every sorting. So merge is to divide each part until it reach to last. And make selecting sort. So it is fast. And customers doesn't wait much time while make sorting.
Bubble sort	For staff data, because it can have less data of staff in the text file. So for less data I will use bubble sort for the staff and if the new staff will add it doesn't have any problem. It can sort easily. As staff can have only less than 30 in library bubble sort is suitable. Because of less data even new staff is arrived it can be 1 or 2. Not too much data.
Binary Search	For both books and customer data I will use jump search because it divide all the array with half and jump and use liner so. Performance is good enough for large data. Because it is suitable with both sort. It need sorting array to search. So quick sort and merge sort are sorted and easy to apply jump search. And jump search is divide the array and check the number less than or grater. So it can only have left and right data. So searching is fast enough both book list and customer data.
Linear Search	As staff data are less I want to use linear search. When data is less, linear search can search easily. Main reason is staff data is less. It doesn't need any sorting data to search because it check with each room with keyword. And staff data is less so searching process is fast.

Recursive Algorithm

Recursive algorithm is a kind of method which calls itself directly or indirectly the corresponding is called as recursive algorithm. It make a small value of input value. It use to solve same problem with same solution. It is a kind of repeat process for the single process. It can use for a problem which can't solve with looping. Performance of recursive is slow because it called itself. Recursive algorithm is use in the complex process problem and it is easy to understand but the performance is not good enough. To use recursive it is very careful to use although it is easy to understand. Because if it wrong usage, it can happen memory stark over flow. Or if the base case is wrong it can have problem with the recursive algorithm.

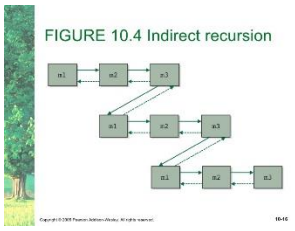
In recursive algorithm it has direct recursive and indirect recursive which called itself directly or go through other methods. Direct recursion is a method which invokes itself and direct recursive is the expression is same in the first paragraph. So here is an example of direct recursive algorithm.

```
public static void main(String[] args) {
  // TODO code application logic here
  printHello(5);

}

public static void printHello(int count)
{
  if(count>=1)
  {
    System.out.println("Hello");
    printHello(count-1);
  }
}
```

In photo, you will see printHello method with parameter. If the count is less than equal to 1 it will print "Hello". You will see the count-1 in it's method. It will reduce count until reach 1 which have the total count in main method. In printHello method. It call itself by reducing count. So it is a directly call is called direct recursive algorithm.

Indirect recursive opposite of direct recursive algorithm. Indirect recursion is when method called form a different method and its turn called the original method. It use more than one method. Indirect it only use one method itself. Indirect recursive called more than one method and use recursively.

A good example is directory traversal program. One method for navigating the hierarchy and one for processing the files. If the file turns out to be a directory then the original process folder method is called and so on.



When use Recursive Algorithm?

It use when the process is huge or big and to see the code clearly and not to have much tracking trace for the program. Like quick sort, merge sort. In quick sort, you have to make partition, swap process, and check condition process. So if you make one layer code, you will find too much nervous and you will have difficult in tracing. So generally recursive algorithm is used for complex and clear process. So to clear quick sort and easy to trace recursive algorithm can use in quick sort. Not only quick. It is up to the condition of the program. If you want to have clear and to solve complex condition. You should use quick sorting. But

recursive code doesn't use in every condition. If a program need to have clear to understand and complex condition, recursive algorithm can use.

So here is an example with quick sort. You can have code in one layer of quick sort. The main difference is too difficult track and very complex code and it is difficult to understand. First we have to make a partition of quick sort array and get pivot. So you will have a partition method for process. After partition you will get pivot and make quick sort. In this case I want to show how recursive work. So we will do quick sort for the recursively.

```
void quickSort(int arr[], int low, int high)
{
    if (low < high)
    {
        /* pi is partitioning index, arr[p] is now
           at right place */
        int pi = partition(arr, low, high);

        // Separately sort elements before
        // partition and after partition
        quickSort(arr, low, pi - 1);
        quickSort(arr, pi + 1, high);
    }
}
```

Recursive use parameter. So you will see a quick sort method. In this case you will see the calling method of quick sort. Parameter data in the quick sort method are recursively called. And data will be transmitted to itself. If other method called that method. Parameter data will be inserted. So in this case, you will clear and easy to find tracking.

How to use Recursive Algorithm?

As I says recursive algorithm is use to solve for complex condition which can't solve with looping. So it is use in complex case. So how to use it? Called it's method in it's scope. It is a little bit complex. Before you use recursive algorithm you should think about the data structure with steps by steps by using pseudo code or flow chat by thinking carefully and tracing each steps. Because if you make wrong recursive, it can have memory stack overflow and that problem is doesn't same with other program problems, it can harm to your device or memory. To use recursive, you should have to think about the parameter, base state which make to stop the sequence or loop and call it's method to do again and again. If you are wrong in base case, you will face with memory over flow or which relate with that problems. Think about the condition steps by steps to prevent from memory stack problem.

Why use recursive Algorithm?

As I says, it is to solve for the complex problems. Recursion can typically allow to use with other concepts such as immutability. Mostly for the looping which is used to be altering a mutable data structure. Using immutable data structure can allow for things like simple parallelizing of your functions. So you can distribute the work among several CPU's without running the risk of corrupting your tasteful object. If you

use recursive the performance is not good enough because of the memory over problems. It just need to use for the complex process for tracking easy for the programmer.

Example Usage of Recursive Algorithm

So as an example, mostly use in algorithm process. It can be in hacking, hashing, encryption. Because they do repeat and repeat same process for all. In hacking to lock the hashing code they should use recursive algorithm to apply. So sometimes hacking apps are memory over problem for hackers. Because simply base case is wrong for hacking algorithm. Hacker can't extract and doesn't know about the Company A's hashing. So they use that kind of recursive algorithm to think about the hashing design of the Company A. And to solve heavy problems to subs problems.

Recursive in Merge sort

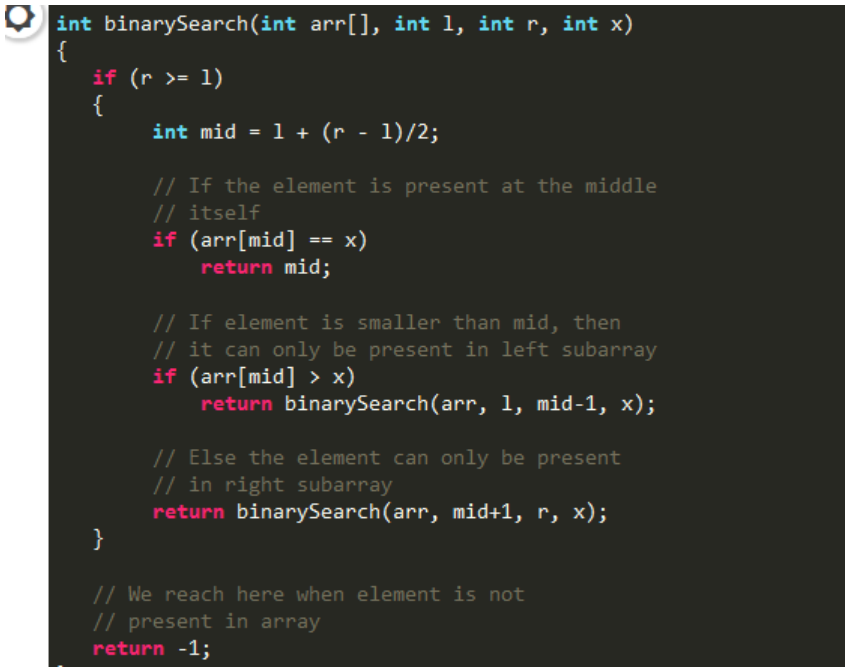
We already know that merge sort is divide until the room doesn't reach to the last. And use selecting sort to make merge sort. So in merge it will have many steps if you write code in one layer, you will have difficult to trace of marge sort. So you have to use different methods and recursive algorithm to apply merge sort. So here is an example.

```
void mergeSort(int arr[], int l, int r)
{
    if (l < r)
    {
        // Same as (l+r)/2, but avoids overflow for
        // large l and h
        int m = l+(r-l)/2;

        // Sort first and second halves
        mergeSort(arr, l, m);
        mergeSort(arr, m+1, r);

        merge(arr, l, m, r);
    }
}
```

So you will see mergesort method and it's called it's self with parameter. That parameter form he main method and form partition method to put it. Those data will transfer to the partition method to make condition of the sorting. With the parameter, partition will take out and sort method will do sort in main method.

Recursive in Binary Search

Binary also need to apply recursive algorithm. Binary is divided and make condition to get the data. It will have many steps to finish and it seem easy in outside. But with the code it can complex for the programmer. So recursive algorithm can help programmer to trace easily.



```
int binarySearch(int arr[], int l, int r, int x)
{
   if (r >= l)
   {
        int mid = l + (r - l)/2;

        // If the element is present at the middle
        // itself
        if (arr[mid] == x)
            return mid;

        // If element is smaller than mid, then
        // it can only be present in left subarray
        if (arr[mid] > x)
            return binarySearch(arr, l, mid-1, x);

        // Else the element can only be present
        // in right subarray
        return binarySearch(arr, mid+1, r, x);
   }

   // We reach here when element is not
   // present in array
   return -1;
```

You will see binarySearch method and it's called itself. it find middle number from left and right. Those left and right will get form the parameter. And parameter data will come from main method. This method is also called in main method to get data and make sort. And it returns it's method and -1. If -1 it can't find. Search data doesn't have in the array list. If its returns it will make binary search up to the condition.

In those case, in merge sort and binary search. It can easily trace all the program. It doesn't have any performance for the program. If the parameter is wrong, it can only have memory stack over flow problems. So think careful for the base to prevent from problems. I hope you will understand about recursive algorithm.

********** Thank You **********