**ALGORITHM**

Algorithm is the method or the steps that we have to follow in order of finding and slove a problem , And Searching problem is one of the big famous problem which is searching for an information among thousands of informations Example : Google , you search for a thing Among billions of information and with all that google dones’t take more that 1s to find you order and sure google doesn’t search in all informations but it’s use a good algorithm for bring your order . and for well inderstand this , imagine that you have an aray :

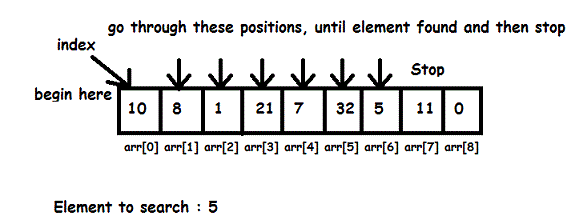
4|8|9|1|6|0|8|7|2|3

The computer isn’t like us , the computer can’t see all index inputs but can see only array inputs that it’s in it and the computer see the array index by index example if the computer in index 4 so computer see only index 4 and for him he is blind ( اعمى) for other inputs , and know you have array [1000] so for search in this inputs you have many methods or many algorthims , the first is searching in this array 1 by 1 which mean linear search algrothim and best scenario of this algorithm is finding the element from the first step and wort scenario of this algorthm is finding the element at the end of the array :

**O (big o) 🡺** mean the worst scenario

**big Omega(Ω) 🡺** mean the best scenario

When you code a program you to based on ‘O’ or the worst scenario bacause the best scenarion doesn’t happen evrytime and for make a good program



For make our speech logique we have convert this algrorithm to a reel code , but affter that you have to order you idea in you mind so you have to do a pseudo-code

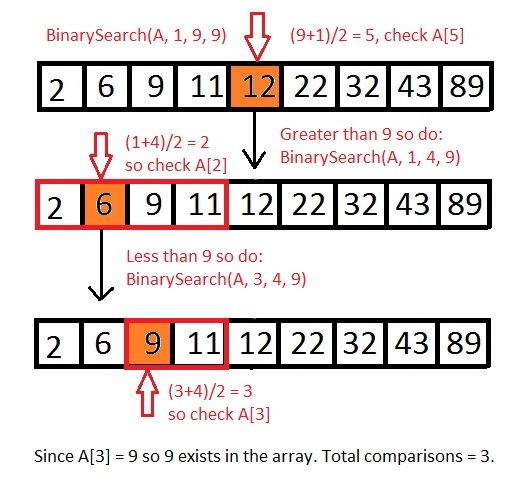
**pseudo-code : order the steps in a small (write first i have to … then … then .. for have … then .. ext ) pseudo-code is you ideas that you have to work on .**

***for i from 0 to n-1***

***if number behind doors [i]***

***Return true***

***Return false***

**But Know imagine with me that you hava an array that contain ordered numbers , so here i don’t think that linear search algorithm is a good choice , But good choice here is Binary search , as the image show**

***if no doors***

***return false***

***if number behind doors[middle]***

***return turn***

***else if number < doors [middle]***

***search from doors[0] to doors[middle -1 ]***

***else if number > doors [middle]***

***search from [middle+1] to n-1***

**Linear Search**

**All content of this episode , you will find it in practice files**

**NOTE : We can’t compare two strings**

**Strcasecmp – compare two strings ignoring case**

**Strcmp – compare two strings case-sensitively**

**In the declaration of an array , if you will fill array directly it not necessary to declare the lenght of the array**

**String s[] = {15,5,6,8,3,1,0,7,8,9} ;**

**But if you will fill array indirectly so it necessary to specify the lenght of the array , bacause you will just book or reserve a place in memory**

**Phonebook**

**During making phonebook program , we had to create a new data type or a new struct , and this struct contain the name (string) and number (string and not int because we don’t do any operation on it ) , for do that , we write berfore main function :**

 struct  phonebook{

           string name ;

           string number ;

           int age ;

      };

**And we have used this struct for craft an array :**

 struct phonebook people[2];

**for abbreviation :**

typedef struct  {

           string name ;

           string number ;

           int age ;

      }pb;

**So now it enought to write just the abbrivaition , to declare this new struct :**

  pb people[2];

**Sorting Algorithms**

**Now we have learned two search algorithm Linear Search and Binary Search but as we know in Binary search the array should be sorted and this is a problem and as we know all problem has a solution , so we have to find an algorithm wich make our array sorted and sure there are many algorithm who do that , and for be honest sorting algorithms are the moset important and the most used ,**

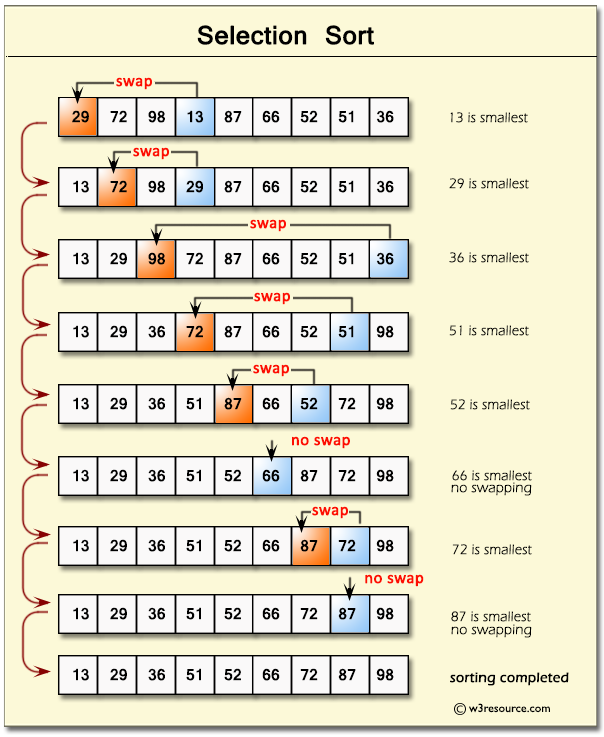
1. **Selection sort :**

**for i from 0 to n-1**

**Find smallest number between numbers[i] and numbers[n-1]**

**if you find numbers small than numbers[i]**

**swap smallest number with numbers [i]**

****

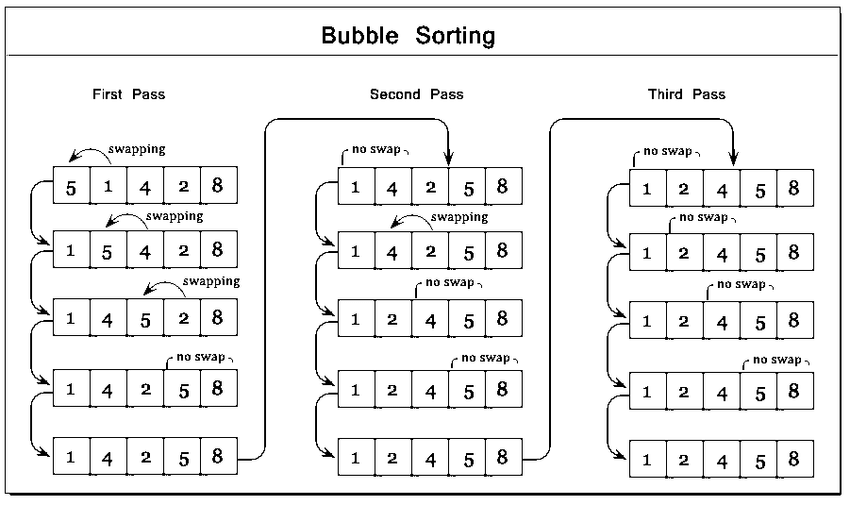
**The big o (O) of this algorithm is n²**

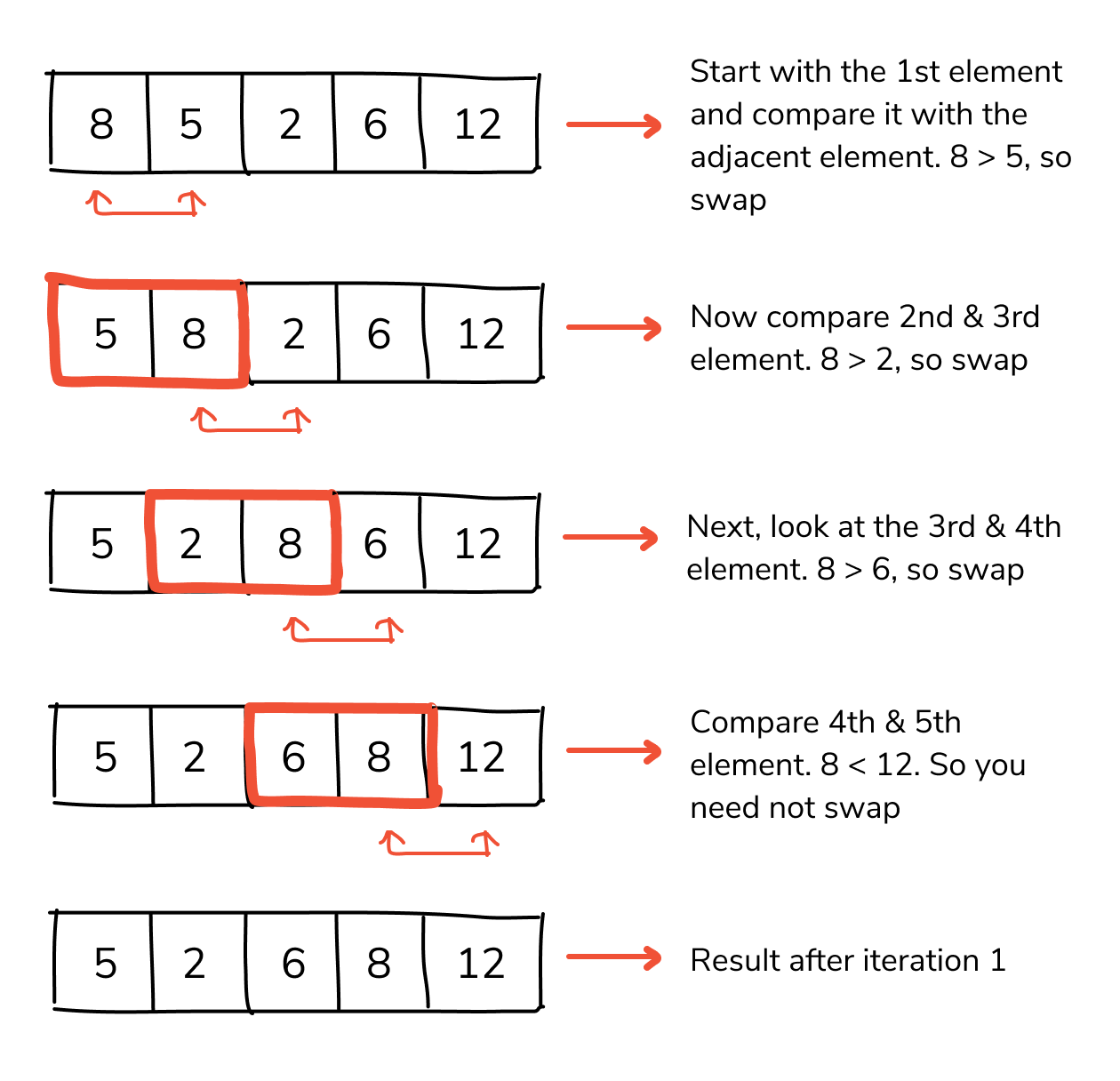
**And also big omega is n² too**

**See The Code of This Algorithm in cs50 practice files**

**2.Bubble sort :**

**So bubble sort is an algorithm wich make our array sorted by comparing two consecutive numbers ,**

****

****

**Pseudo-code :**

***Repeat n-1 times***

***For i from 0 to n-2***

***if numbers [i] and numbers [i+1] out of order***

***swap them***

***if no swaps***

***Quit***