Greedy Approach

Bubble Sort Algorithm

> In bubble sort, each pair of adjacent elements are compared and the elements are swapped if they are not follow the ordering rule.

Example

Let's take an array of 5 elements.

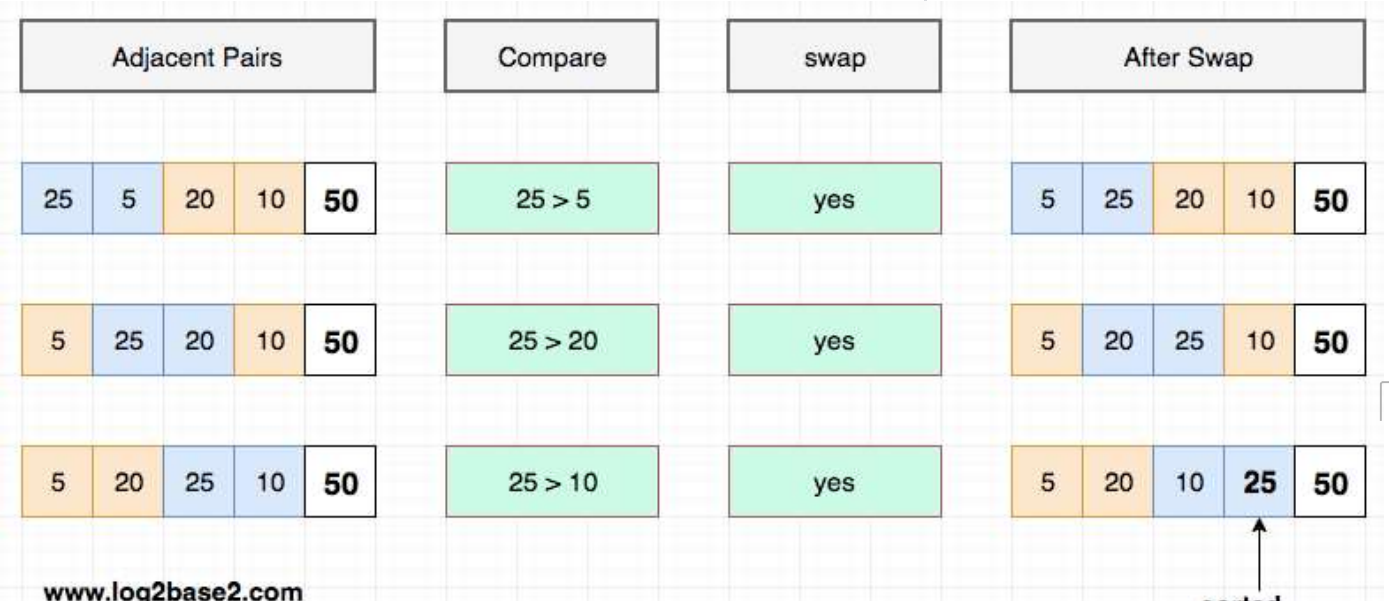
int arr[5] = {50, 25, 5, 20, 10}

Step 1

Algorithms

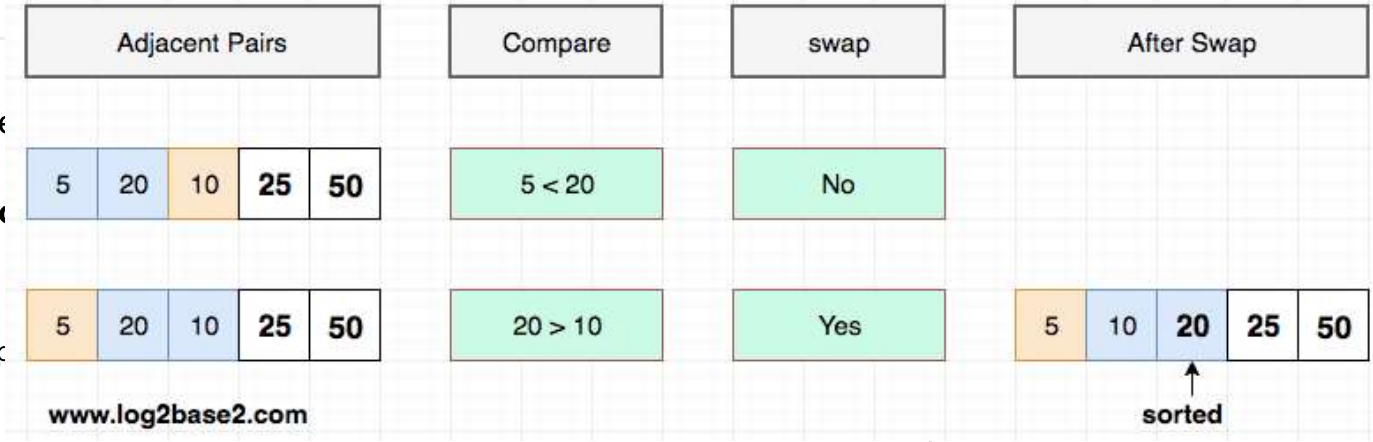
> We can notice that one element has been sorted after the above process.
>
> In general, to sort **N** element using bubble sort, we need to do the same process **N-1** times.
>
> From next iteration onwards, we can skip the sorted elements. i.e. 50

Step 2



Step 3

Algorithms

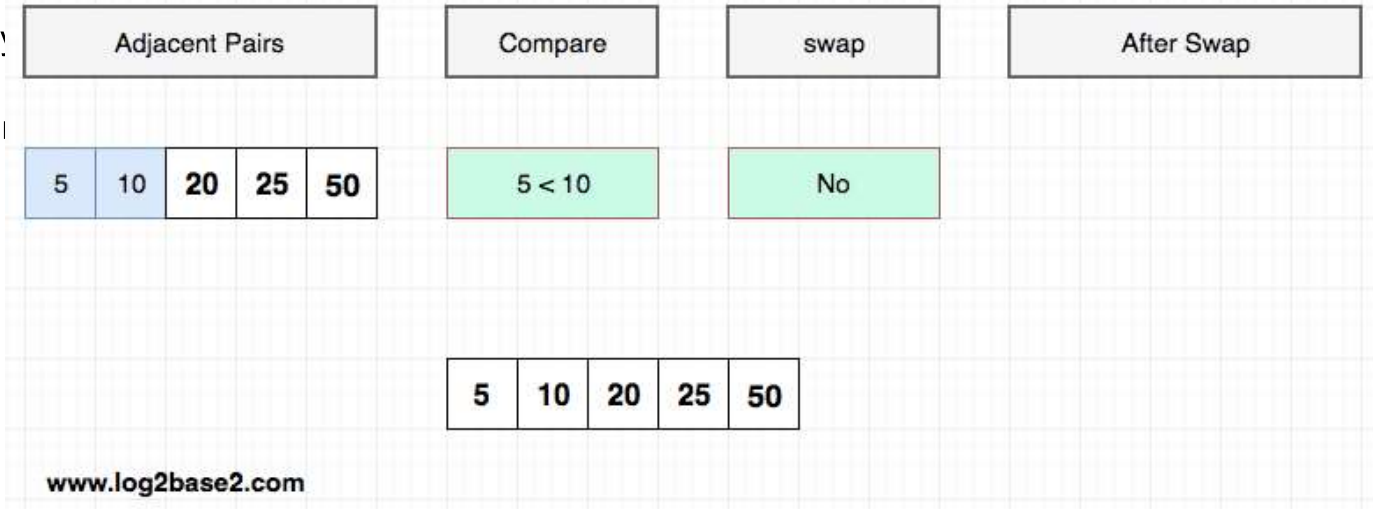


(https://www.log2base2.com/algorithms/sorting/bubble-sort-algorithm-in-c.html)

Step 4

Quicksort

(https://www.log2base2.com/algorithms/sorting/quick-sort.html)



Bubble Sort Program


```
/*
 * Program  : Bubble sort
 * Language : C
 */
#include<stdio.h>
#define size 5

void swap(int *x, int *y)
{
    int temp = *x;
    *x = *y;
    *y = temp;
}

void bubbleSort(int arr[])
{
    int i,j;
    for(i = 0; i < size-1; i++)
    {
        for(j = 0; j < size-1-i; j++)
        {
            if(arr[j] > arr[j+1])
                swap(&arr[j],&arr[j+1]);
        }
    }
}

int main()
{
    int arr[size] = {50, 25, 5, 20, 10}, i;

    bubbleSort(arr);

    printf("After the bubble sort\n");
    for(i = 0; i < size; i++)
        printf("%d ",arr[i]);

    return 0;
}
```

Binary Dollar Question

In the above program, we have used **j < size-1-i** condition in the inner for loop.

```
for(j = 0; j < size-1-i; j++)
```

What change it will bring? Can you guess?

Optimizing the Bubble Sorting Algorithm

We can optimize the bubble sort algorithm further.

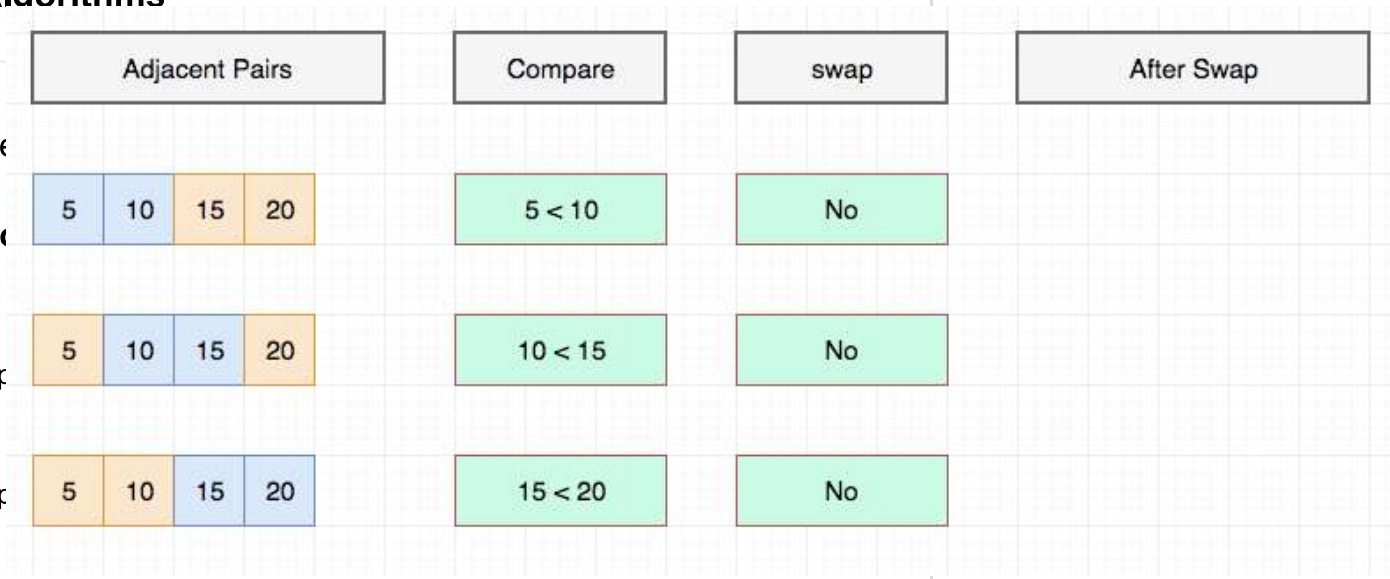
Let's take a sorted array.

Example

int arr[5] = {5, 10, 15}

Bubble sort operation will be like below.

Algorithms



(https://www.log2base2.com/algorithms/sorting/quick-sort.html)

Dynamic Programming

Greedy Approach

Observation

> We can notice that the sorted array doesn't need any swap operation.
>
> So, if no swap operation takes place, we can ensure that the array is sorted and we can break the process.
>
> It will improve the efficiency of the bubble sort algorithm.

Example

X

Algorithms

Searching

Sorting

Selection Sort
(https://www.log2base2.com/algorithms/sorting/selection-sort.html)

Bubble Sort
(https://www.log2base2.com/algorithms/sorting/bubble-sort-algorithm-in-c.html)

Quicksort
(https://www.log2base2.com/algorithms/sorting/quick-sort.html)

Dynamic Programming

Greedy Approach

```
/*
 * Program  : Improved Bubble Sort
 * Language : C
 */
#include<stdio.h>
#define size 5

void swap(int *x, int *y)
{
    int temp = *x;
    *x = *y;
    *y = temp;
}

void bubbleSort(int arr[])
{
    int i,j,flag;

    for(i = 0; i < size-1; i++)
    {
        flag = 0;
        //if any swap operation takes place, set flag as 1
        for(j = 0; j < size-1-i; j++)
        {
            if(arr[j] > arr[j+1])
            {
                swap(&arr[j],&arr[j+1]);
                flag = 1;
            }
        }
        //if flag remains 0, break the process
        if(flag == 0)
            break;
    }
}

int main()
{
    int arr[size] = {50, 25, 5, 20, 10}, i;

    bubbleSort(arr);

    printf("After the bubble sort\n");
    for(i = 0; i < size; i++)
        printf("%d ",arr[i]);
```

```
    return 0;
}
```

Run it (https://www.log2base2.com/algorithms/sorting/try-it-bubble-sort-algorithm-in-c2.html)

PAGE -- (https://www.log2base2.com/algorithms/sorting/selection-sort.html)
(https://www.log2base2.com/algorithms/sorting/bubble-sort-algorithm-in-c.html)

PAGE ++ (https://www.log2base2.com/algorithms/sorting/quick-sort.html)

X