In [ ]:

```
'''
quick sort
merge sort
graph(theory,types)
directed
un-directed
graph traversals(BFS,DFS)
shortest path algos
'''
'''
bubble sort
selection sort
insertion sort
'''
```

In [2]:

```
#selection sort
#a pointer points to a ele,and swaps with smallest ele in the remaining array
# 10 70 40 20 60 50
# 10 20 40 70 60 50
# 10 20 40 70 60 50
# 10 20 40 50 60 70
# 10 20 40 50 60 70

# insertion sort
# merge sort
# quick sort
def qs(arr):
    if len(arr)<=1:
        return arr
    pivot=arr[0]
    lft_arr=[i for i in arr[1:] if i<=pivot]
    rt_arr=[i for i in arr[1:] if i>pivot]
    return qs(lft_arr)+[pivot]+qs(rt_arr)
arr=[30,20,50,40,10,20]
print("arr",arr)
print("sorted arr",qs(arr))
```

```
arr [30, 20, 50, 40, 10, 20]
sorted arr [10, 20, 20, 30, 40, 50]
```

In [3]:

```
# BFS -> Breadth First Search
def BFS(graph,start):
    q=[start]
    visited=[start]
    while len(q)!=0:
        ele=q.pop(0)
        print(ele)
        for i in graph[ele]:
            if i not in visited:
                q.append(i)
                visited.append(i)

graph={
    "a":["b","c"],
    "b":["a","d"],
    "c":["a","d"],
    "d":["c","e","b"],
    "e":["d"]
}
BFS(graph,"c")
```

```
c
a
d
b
e
```

In [8]:

```
# a-b
# b-c
# c-d
# d-e
# find a-e
```

In [ ]:

```
# BFS , visit all possible nodes at the same tym
# DFS, goes only one at a time
```