NS 11 Apr 2023

Main.java

Main.java > Main > main(String[])

```java
// Yogesh
class Main {

    public static int linearSearch(String[] menu, String favourite) {
        for (int i = 0; i < menu.length; i++) {

            if (menu[i] == favourite) {

                return i;

            }
        }
        return -1;
    }

    Run | Debug
    public static void main(String args[]) {

        String[] menu = { "samosa", "dosa", "idli", "coke", "fruiti", "noodles" };

        String favourite = "dosa";

        int index = linearSearch(menu, favourite);

        if (index == -1) {
```

menu.length = 6

menu

s	d	i	c	f	n
0	1	2	3	4	5

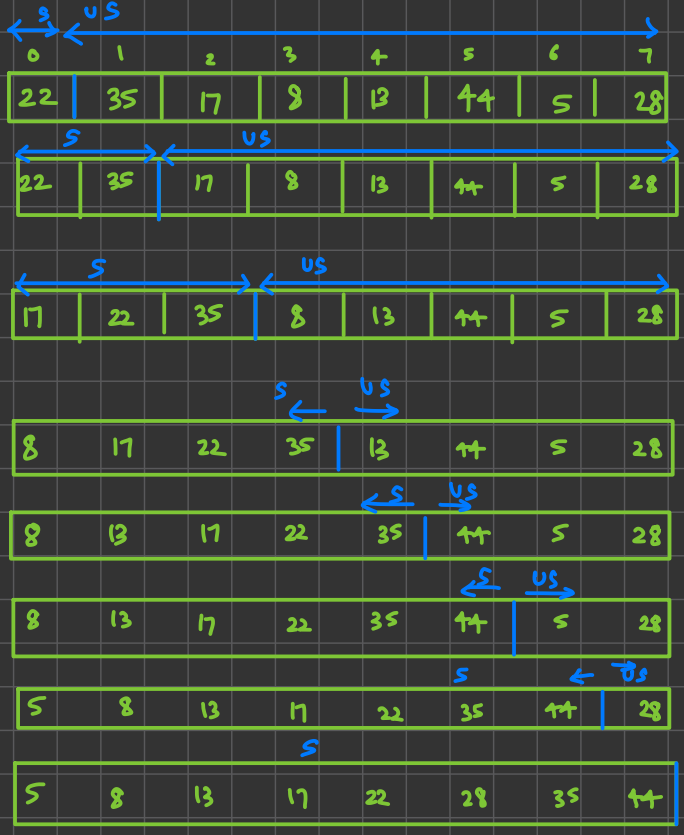
i

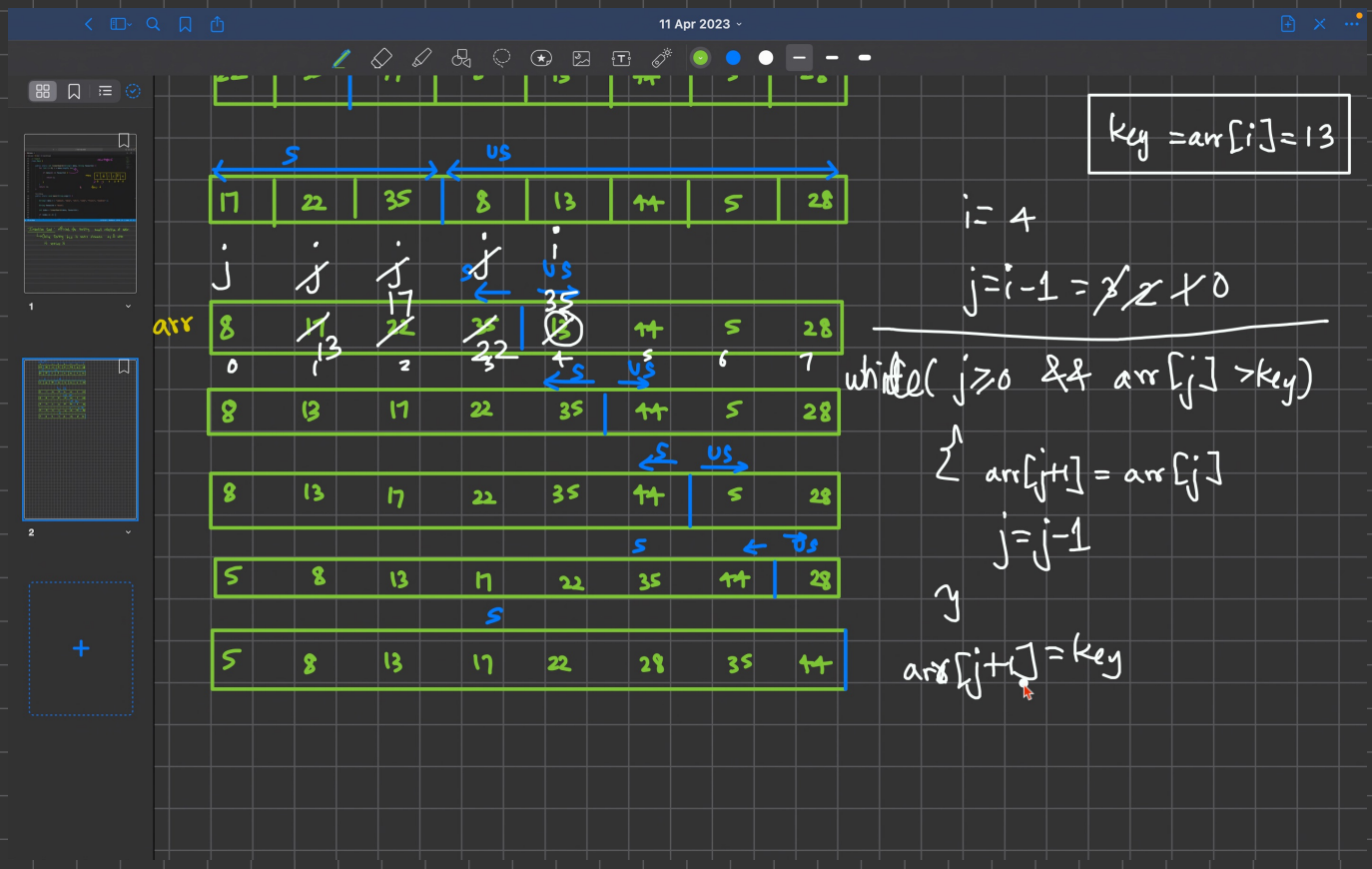
fav = d

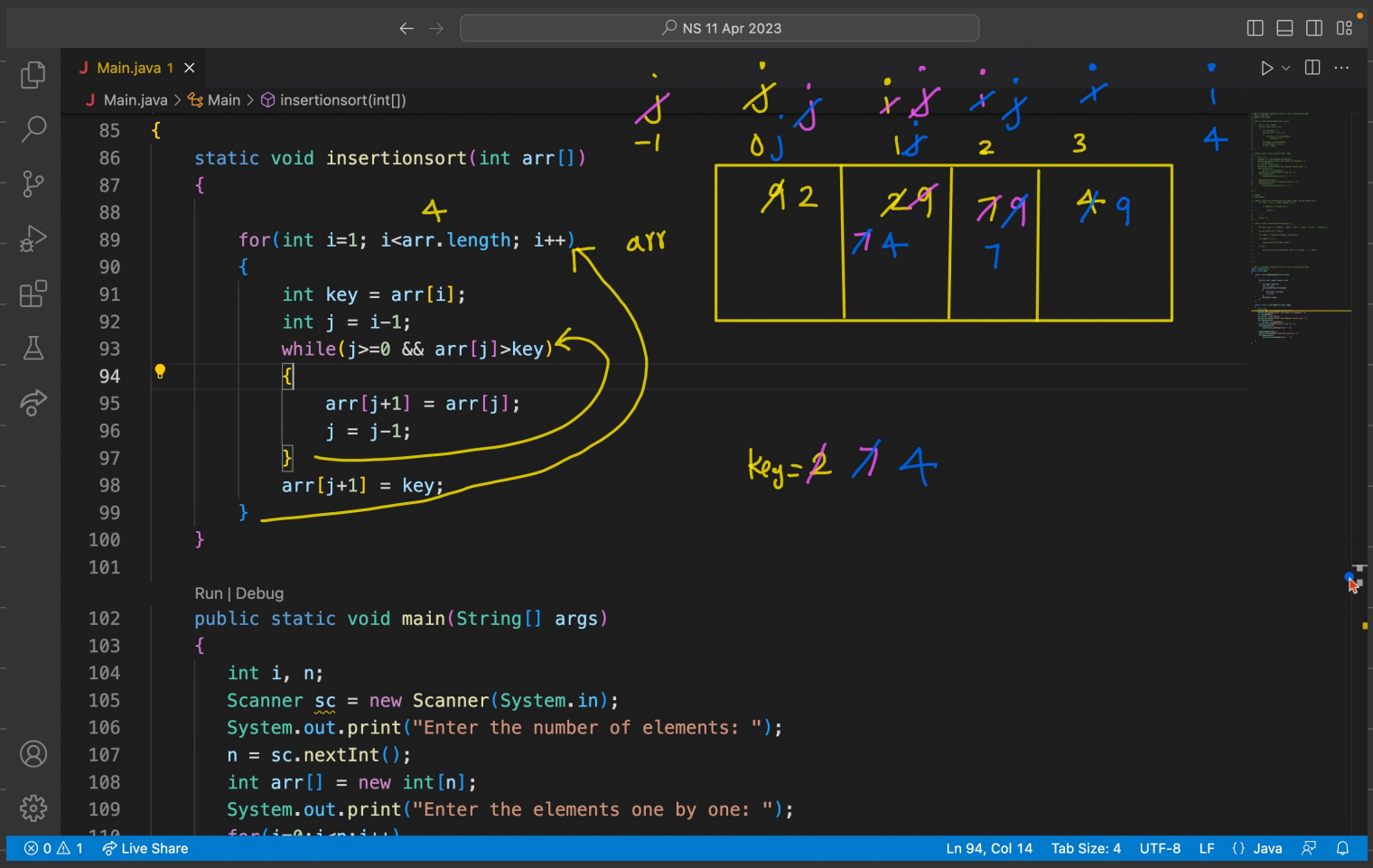
Live Share Ln 70, Col 1 Tab Size: 4 UTF-8 LF Java

Insertion Sort : efficient for sorting small collection of data.

↳ Online Sorting : bcz it sorts elements as & when it receives it.


S
US
0 1 2 3 4 5 6 7
22 35 17 8 13 44 5 28
S
US
22 35 17 8 13 44 5 28
S
US
17 22 35 8 13 44 5 28
S
US
8 17 22 35 13 44 5 28
S
US
8 13 17 22 35 44 5 28
S
US
8 13 17 22 35 44 5 28
S
US
5 8 13 17 22 35 44 28
S
5 8 13 17 22 28 35 44


11 Apr 2023
key = arr[i] = 13
i = 4
j = i - 1 = 3 ≥ 0
while(j ≥ 0 && arr[j] > key)
{
arr[j+1] = arr[j]
j = j - 1
}
arr[j+1] = key
arr
0 1 2 3 4 5 6 7
17 22 35 8 13 44 5 28
8 17 22 35 35 44 5 28
8 13 17 22 35 44 5 28
8 13 17 22 35 44 5 28
5 8 13 17 22 35 44 28
5 8 13 17 22 28 35 44



Quick Sort

pivot = 7

0	1	2	3	4	5	6	7	8
7	6	10	5	9	2	1	15	7
7	6	7	5	9	2	1	15	10
7	6	7	5	1	2	9	15	10
2	6	7	5	1	7	9	15	10

Revise Recursion < Fibonacci, Factorial — if u really wnt to understand Quick Sort.