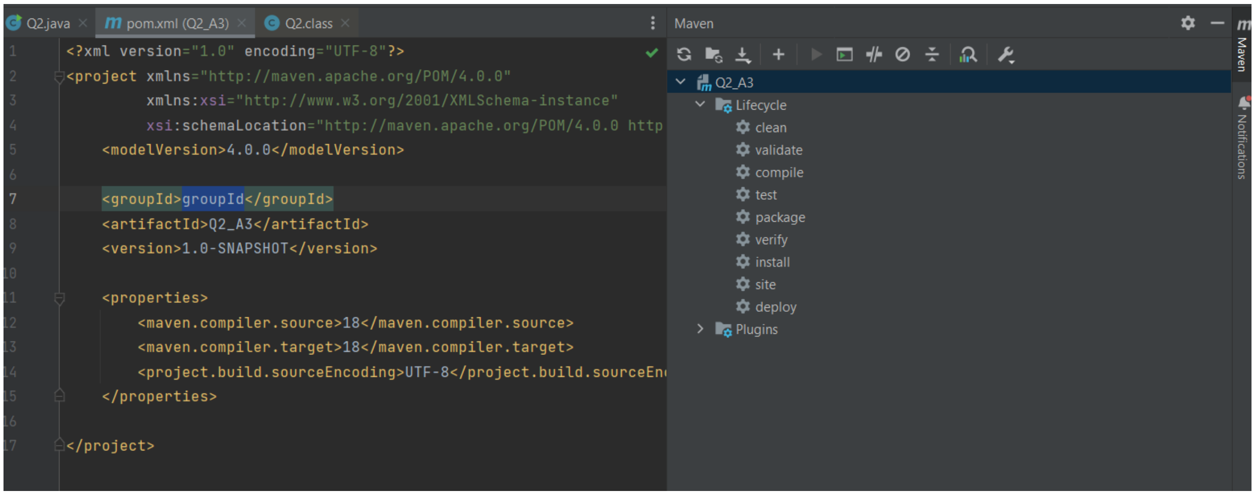
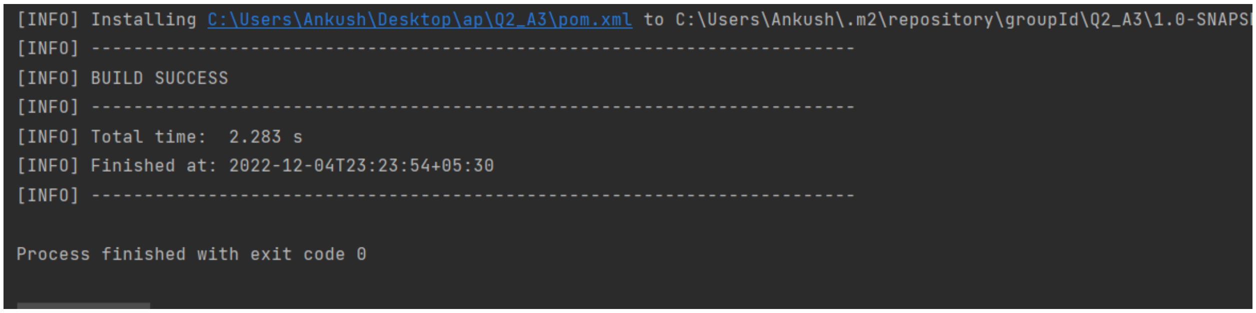
> open maven project in intellij
> open pom.xml file
> click on maven>lifecycycle



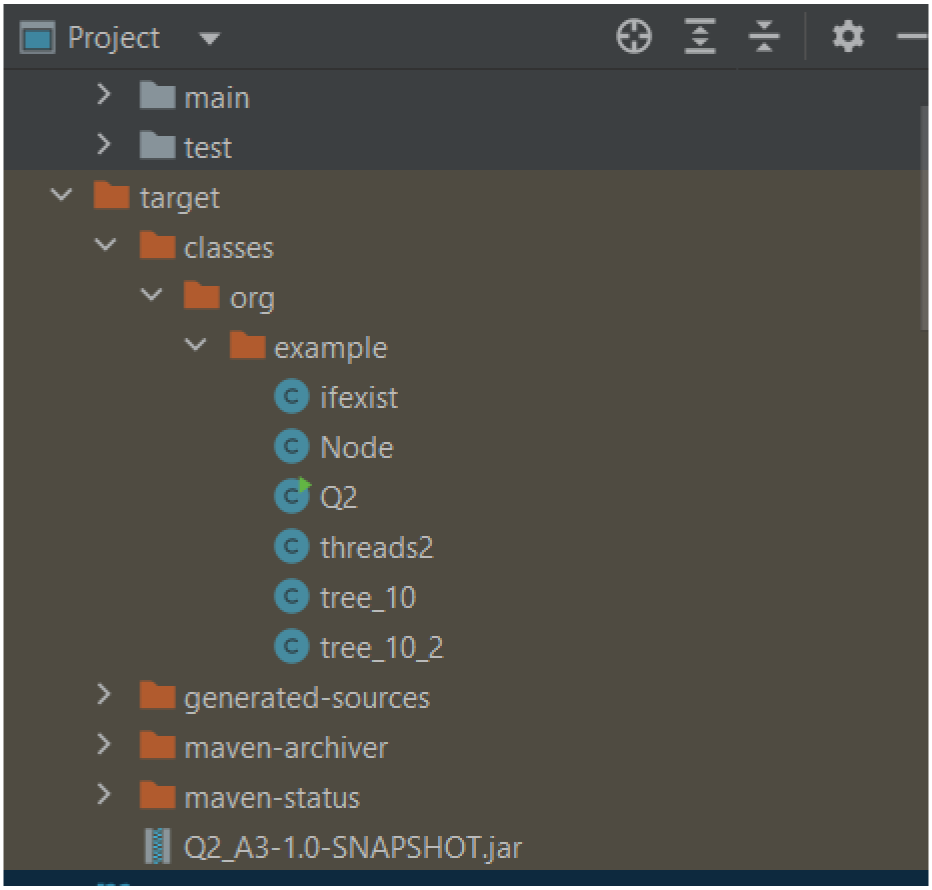
```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http
    <modelVersion>4.0.0</modelVersion>

    <groupId>groupId</groupId>
    <artifactId>Q2_A3</artifactId>
    <version>1.0-SNAPSHOT</version>

    <properties>
        <maven.compiler.source>18</maven.compiler.source>
        <maven.compiler.target>18</maven.compiler.target>
        <project.build.sourceEncoding>UTF-8</project.build.sourceEn
    </properties>

</project>
```

> double right click on install
>Result of this should be like



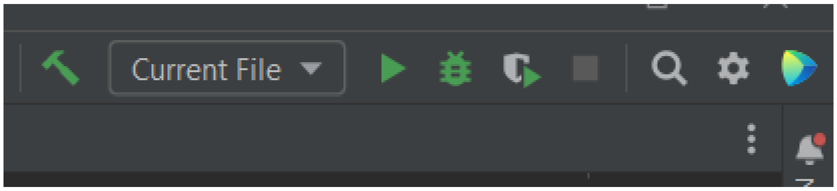
```
[INFO] Installing C:\Users\Ankush\Desktop\ap\Q2_A3\pom.xml to C:\Users\Ankush\.m2\repository\groupId\Q2_A3\1.0-SNAPS
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  2.283 s
[INFO] Finished at: 2022-12-04T23:23:54+05:30
[INFO] ------------------------------------------------------------------------

Process finished with exit code 0
```

> open on Q2.class



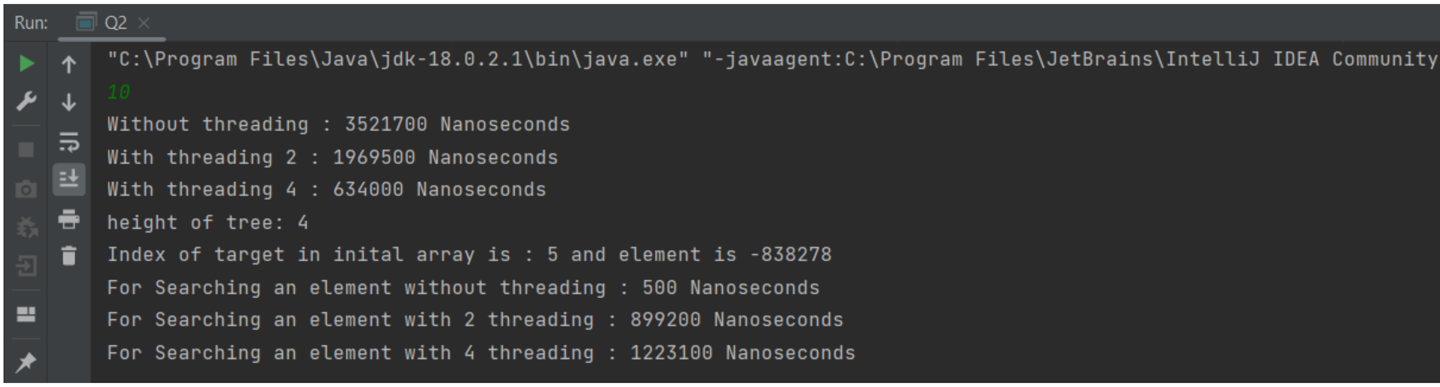
```
Project
    main
    test
  target
    classes
      org
        example
          ifexist
          Node
          Q2
          threads2
          tree_10
          tree_10_2
    generated-sources
    maven-archiver
    maven-status
    Q2_A3-1.0-SNAPSHOT.jar
```

> click on run option in right side of intellij



> input : Give the length of the array

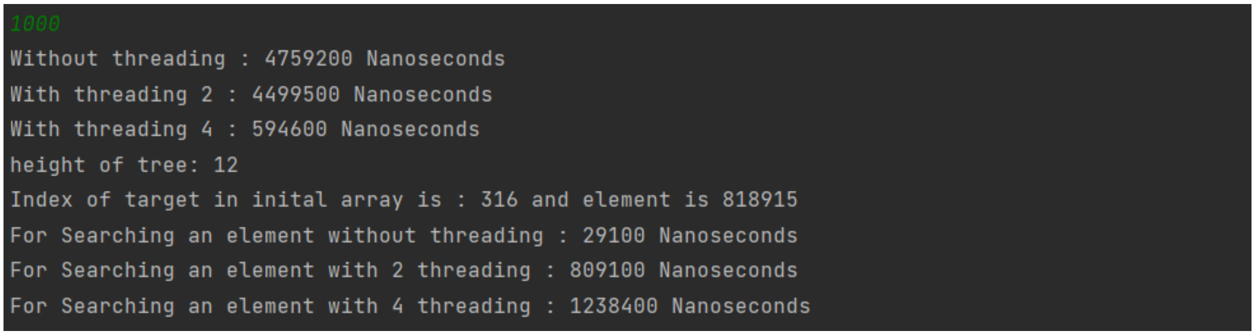
Result:



```
Run:  Q2
"C:\Program Files\Java\jdk-18.0.2.1\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community
10
Without threading : 3521700 Nanoseconds
With threading 2 : 1969500 Nanoseconds
With threading 4 : 634000 Nanoseconds
height of tree: 4
Index of target in inital array is : 5 and element is -838278
For Searching an element without threading : 500 Nanoseconds
For Searching an element with 2 threading : 899200 Nanoseconds
For Searching an element with 4 threading : 1223100 Nanoseconds
```

Input is 10

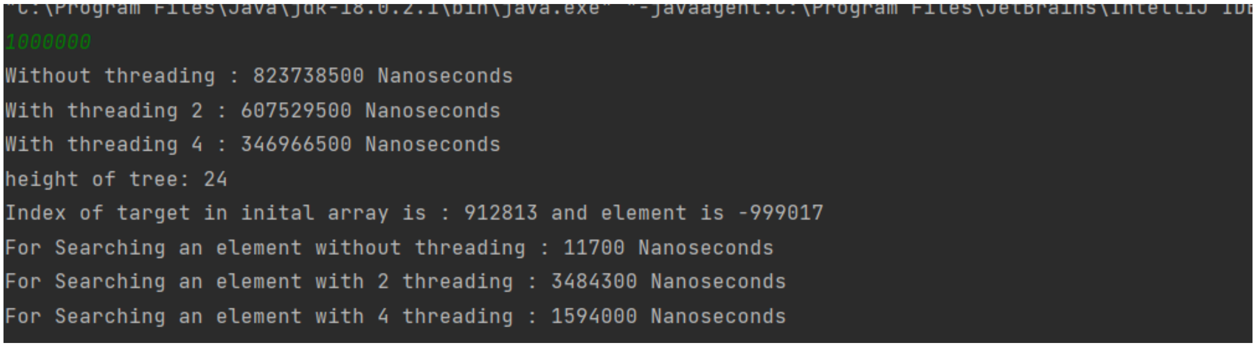
Height is less than or equal to 1.44*log(n) i.e. 4 so the tree is balanced.
Time for searching with threading may come more in some cases because of threading.
I first made preorder transversal, and then I ran threads on the half/quarter length of the array in parallelisation if in preorder element occurs before the first half/quarter, then the method is broken by the break command, but in parallelisation, other threads will keep on running and thus time is more.



```
1000
Without threading : 4759200 Nanoseconds
With threading 2 : 4499500 Nanoseconds
With threading 4 : 594600 Nanoseconds
height of tree: 12
Index of target in inital array is : 316 and element is 818915
For Searching an element without threading : 29100 Nanoseconds
For Searching an element with 2 threading : 809100 Nanoseconds
For Searching an element with 4 threading : 1238400 Nanoseconds
```

Input is 1000

Height is less than or equal to 1.44*log(n) i.e. 14 so the tree is balanced.
Time for searching with threading may come more in some cases because of threading.
I first made preorder transversal, and then I ran threads on the half/quarter length of the array in parallelisation if in preorder element occurs before the first half/quarter, then the method is broken by the break command, but in parallelisation, other threads will keep on running and thus time is more.



```
"C:\Program Files\Java\jdk-18.0.2.1\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDE
1000000
Without threading : 823738500 Nanoseconds
With threading 2 : 607529500 Nanoseconds
With threading 4 : 346966500 Nanoseconds
height of tree: 24
Index of target in inital array is : 912813 and element is -999017
For Searching an element without threading : 11700 Nanoseconds
For Searching an element with 2 threading : 3484300 Nanoseconds
For Searching an element with 4 threading : 1594000 Nanoseconds
```

Input is 10^6

Height is less than or equal to 1.44*log(n) i.e. 28 so the tree is balanced.
Time for searching with threading may come more in some cases because of threading.

I first made preorder transversal, and then I ran threads on the half/quarter length of the array in parallelisation if in preorder element occurs before the first half/quarter, then the method is broken by the break command, but in parallelisation, other threads will keep on running and thus time is more.