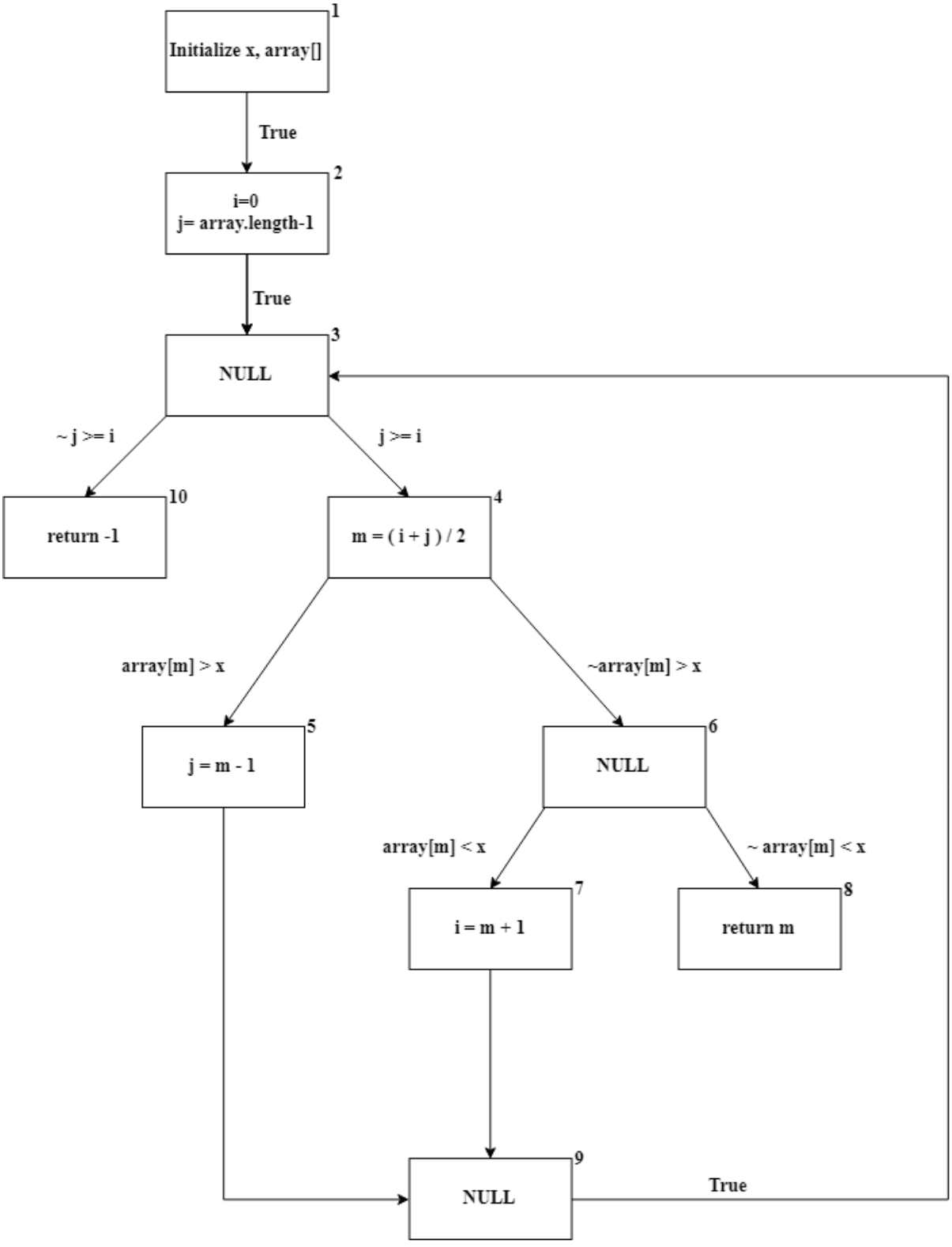
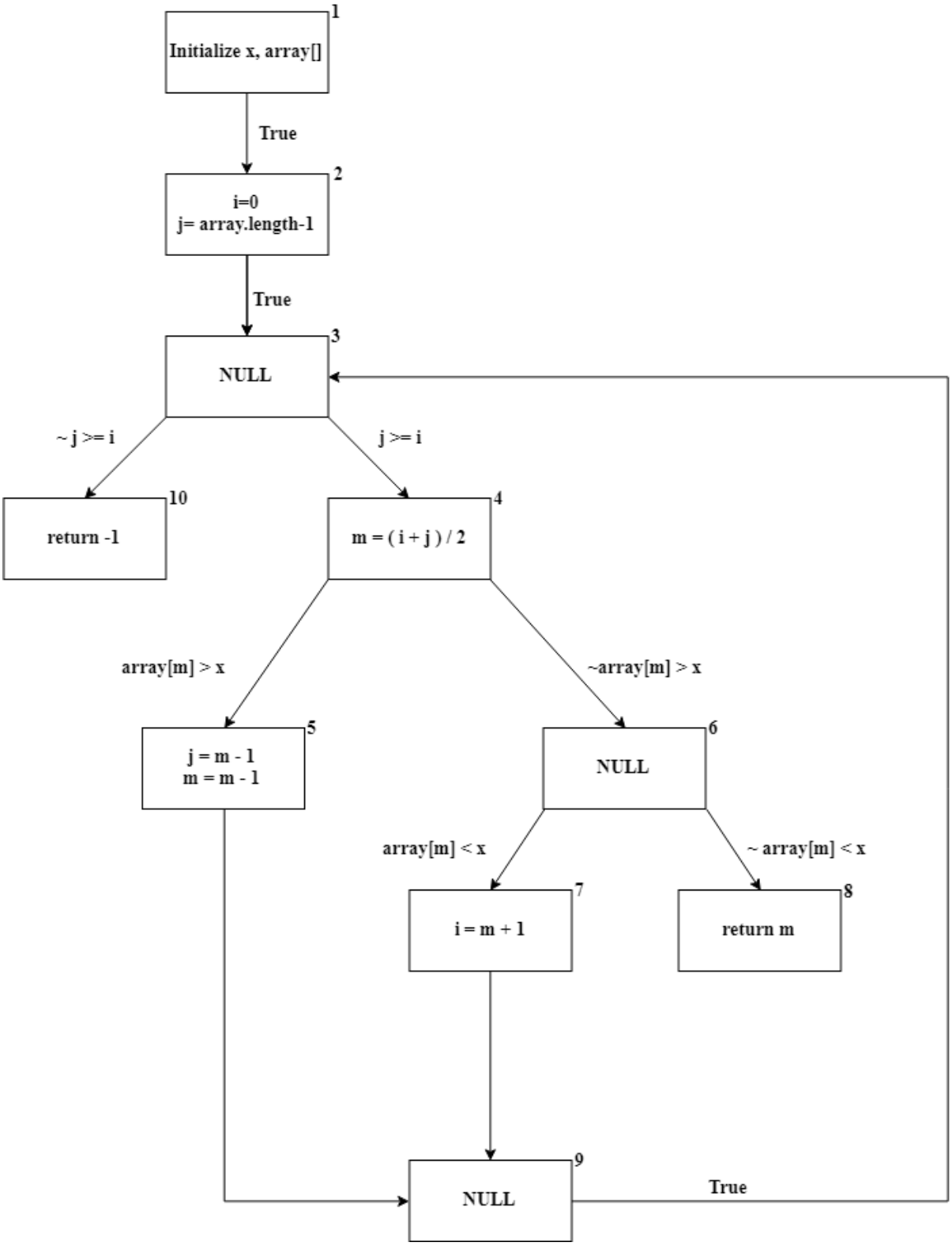
CENG437 - Software Quality Management Homework#3



Data Flow Graph For *binarySearch1*



Data Flow Graph For *binarySearch2*

2-) If array [] has at least one elements in it, path 1-2-3-10 is an infeasible path.

3-) M: u -> d -> d. In side of while m is defined with “m = (i+j) / 2”. In same while part in side of if statement m is assign to again “m = m -1”. In this part m is not used again and caused “dd” data flow anomaly.

4-) The path “1-2-3-4-5-9-3-4-6-7-9-3-4-6-8” is a complete path that satisfies all-defs selection criteria with respect to variable “m”. Paths “4-5”, “4-7”, “4-8” are definition clear path form binarySearch1 ().

5-) The path “1-2-3-4-6-7-9-3-4-5-9-3-4-6-8” is a complete path that satisfies all-defs selection criteria with respect to variable “j”. Path “2-5” is definition clear path form binarySearch1 ().

6-) The path “1-2-3-4-6-7-9-3-4-6-8” is a complete path that satisfies all-defs selection criteria with respect to variable “m”. Paths “4-8”, “4-9” are definition clear path form binarySearch2 ().

7-)

One of the simplest example of DD anomaly;

```
int i = 5;
i = 6;
System.out.print(i);
```

In here “i” is defined and assign with integer 5. After that line “i” is assign 6. “i = 5” equation is not used and assign another value.

One of the simplest example of DU anomaly;

```
int i = 5;
System.out.print(i);
i = 2;
```

In here “i” assign with 2 and do not used. First “i” is defined as 5 and referenced than it defined as 2 and program finished.

One of the simplest example of UR anomaly;

```
int i = 5;
System.out.print(u);
```

In here “u” not even defined but try to reference it.

Ahmet Eroğlu

210201010