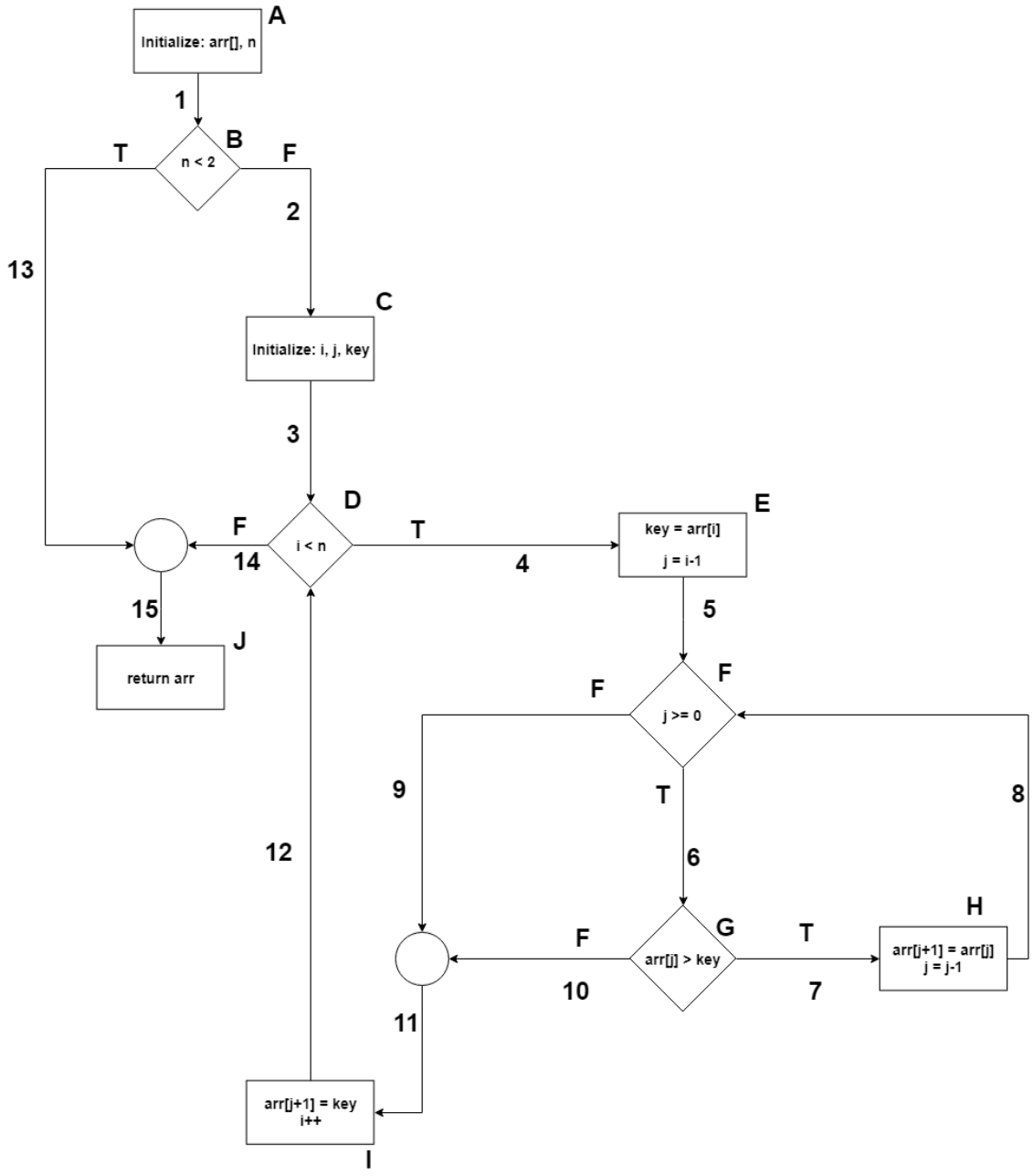
CENG437 - Software Quality Management Homework#2

1-)



2-) There is a path that satisfy complete statement coverage which is:

A-B(F)-C-D(T)-E-F(T)-G(T)-H-F(F)-I-D(F)-J[statements] (1-2-3-4-5-6-7-8-9-11-12-14-15)

3-) If we add two more paths we can satisfy compete branch coverage. This pats are:

A-B (T)-J (1-13-15)

A-B(F)-C-D(T)-E-F(T)-G(F)-I-D(F)-J (1-2-3-4-5-6-10-11-12-14-15)

4-) Path Predicate expressions for;

First path; N<2: false, I<n: true, J => 0: true, arr [I] > key: true, J > 0: false, I<n: false

Second path; N<2: true

Third path; N<2: false, I<n: true, J => 0: true, arr [I] > key: false, I<n: false

5-) Path predicate expressions to generate test input and compute the corresponding are;

First path; Arr [] = {3, 2}, N=2, Expected Output = {2, 3}

Second path; Arr [] = {3}, N=1, Expected Output = {3}

Third path; Arr [] = {2, 3}, N = 2, Expected Output = {2, 3}

6-) There exists a path which is “1-2-3-4-5-9-11-12-14-15”. This path is not feasible because in the first iteration “j” will be never “smaller than 0”. (A-B(F)-C-D(T)-E-F(F)-I-D(F)-J) [Statements]

7-) If we can change the statement “j >= 0” to “j >= 1” it passes 2 of the 3 test cases.

8-) We test our program with 3 different test cases if we add more cases to test maybe we can detect it. My input set is consist of 2 lengths of arrays. If we test our program with larger array size maybe we can detect it. Another way is after all test cases past, we can comparison the hard code and control flow graph of that code, in this way, the faults maybe detect.

Ahmet Eroğlu
210201010