Software Testing: Tutorial 3

Data Flow Testing

Consider the following program:

```
static int find (int list[], int n, int key)
{
    // binary search of ordered list
    int lo = 0, mid;
    int hi = n - 1;
    int result = -1;
    while ((hi >= lo) && (result == -1)) {
        mid = (lo + hi) / 2;
        if (list[mid] == key)
            result = mid;
        else if (list[mid] > key)
            hi = mid - 1;
        else // list[mid] < key
            lo = mid + 1;
    }
    return result;
}
```

This is not a particularly good example of programming but it is useful for the purposes of this tutorial.

- **Prerequisites:** Review the material on Data-Flow based testing in Lectures 7 and 8 and the paper by Frankl and Weyuker.

- **Preparation:** Review the code above; please try to ensure you understand the method and the particular implementation. It is an implementation of binary search of an ordered array.
Activities

1. (10 Minutes) First individually construct the flow graph corresponding to this program.

2. (5 Minutes) Find a partner to work with in the group and check that you agree on the structure of the flow-graph for the program.

3. (10 minutes) For each variable, write down the \(< D, U >\) pairs.

4. (10 minutes) Write down tests that satisfy one of the following coverage criteria:
   
   (a) All \(< D, U >\) pairs
   
   (b) All \(< D, U >\) paths

5. (10 minutes) As a whole class, compare the tests sets devised for the two coverage criteria and discuss which is stronger. Can you think of a test that passes one of the two criteria but fails the other?