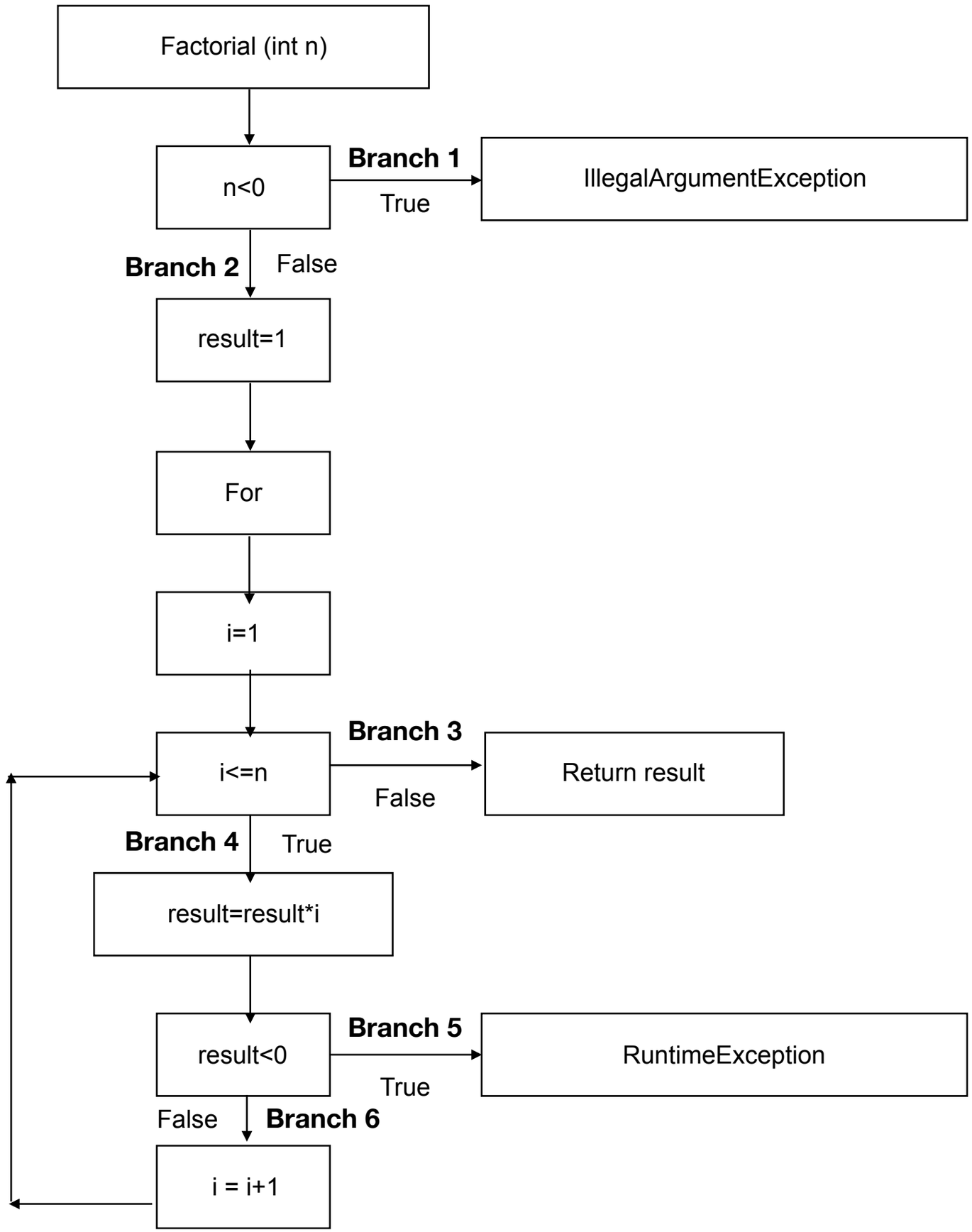
CS 5500: Managing Software Development
Homework 3

Shashwat Sanghavi

Q1: CFG for factorial



B). Test suit for 100% branch coverage

No	Input	Output	Note
1	n=-1	IllegalArgumentException	
2	n=0	1	Boundary
3	n=1	1	
4	n=16	2004189184	Boundary
5	n=17	RuntimeException	Overflow

Explanation

- In the above mentioned test specification, test No 1 checks for negative input. It should provide illegalArgumentException. Covers branch 1.

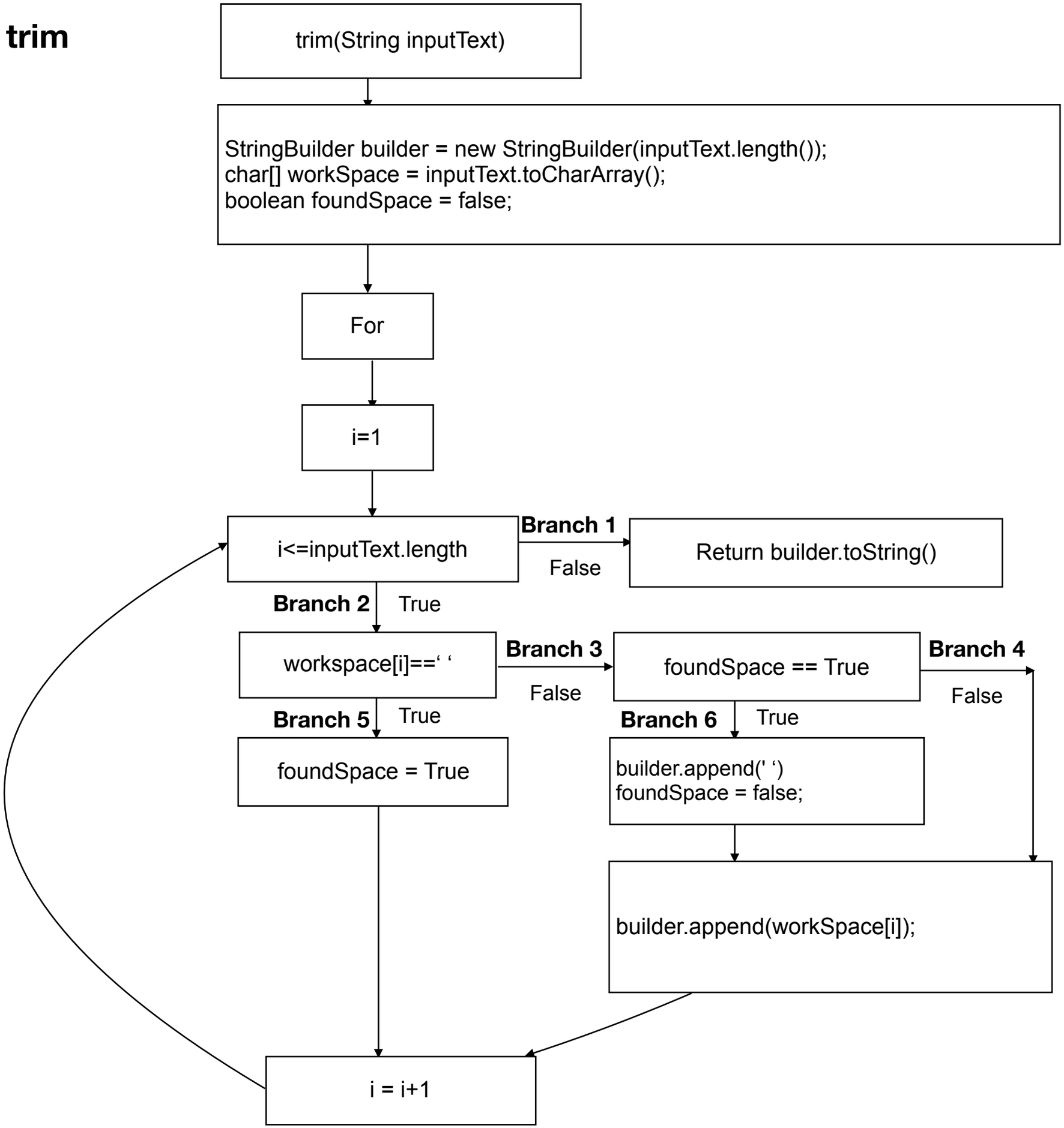
- Test 2 is a boundary condition and should return 1 as it won't go inside the for loop body. Covers branch 2,3.

- Test 3,4 are cases where for loop body would be executed and output would be returned. Covers branch 2,4,6,3.

- In case 5, 17! Is larger than the maximum possible integer and thus overflow should happen. Thus a RuntimeException should occur. Covers branch 2,4,6,5.

- Thus, the above test suite covers all six branches of the CFG and thus it gives 100% branch coverage.

Q2: CFG for trim



B). Test suit for 100% branch coverage

No	InputText	Output
1	“”	“”
2	“ ”	““
3	“ “	““
4	“Hello world”	“Hello world”
5	“hello world”	“hello world”
6	“hello world eferwewer”	“hello world eferwewer”
7	“hello”	“hello”
8	“hello “	“hello”

Explanation

- As in test 1, for an empty input string, the length of string is less than initial value of I and thus it should return an empty string. This case covers branch 1.

- In case of a single space, the for loop would be executed and it will cover branch 2,5 and 1. It should return an empty string as an output.

- In case of multiple spaces, it should return an empty string by following branch 2,5 and 1.

- In test 4, it should return exact same string as there are no occurrences of multiple spaces. It follows branch 2,3,4,5,6,1.

- Thus, the above test suite covers all six branches of the CFG and thus it gives 100% branch coverage.

Q3 which version is better for structural testing?

- IRS_Alternate.youOwe2 <Later one> is better in terms of structural perspective.

- Reason: Calculating income tax involves critical calculations and a miscalculation in a single condition can result into critical error in the tax calculated. Thus for this kind of a software, MC/DC testing should be used.

- For effective MC/DC testing, it would be easier if condition complexity is exchanged with the branch complexity.

- In the later option, the CFG has more branches however the conditions for each branch are simpler than the former one.

- Thus the second approach is a better approach.