

Problema da Janela

Particionamento do Espaço de Entrada

Turma 2015-2 da disciplina MAB640

Projeto de testes

Características consideradas e particionamento para cada característica

1. Para cada parâmetro (F1, F2, F3): seus valores (inteiros) vão de 0 (inclusive) a 400 (inclusive). Característica para F_i : posição da janela no eixo 0..400 (só incluindo valores válidos)
 - a. bloco1: $F_i = 0$,
 - b. bloco2: $0 < F_i < 400$ e
 - c. bloco3: $F_i = 400$
2. Característica (baseada em funcionalidade): número de aberturas
 - a. 0 (janela fechada)
 - b. 1
 - c. 2
 - d. 3

Critério de Combinação

Each Choice

Requisitos

$TREC = \{b_{11}, b_{21}, b_{31}, b_{12}, b_{22}, b_{32}, b_{13}, b_{23}, b_{33}, 0a, 1a, 2a, 3a\}$,

onde b_{ij} significa bloco i da folha j e $0a$ representa zero aberturas, $1a$, uma abertura, etc.

Casos de Teste para a satisfação dos requisitos

- CT1: 0, 0, 0 — satisfaz requisitos: $b_{11}, b_{12}, b_{13}, 1a$
resultado esperado: $400 \times 100 = 40.000$
- CT2: 100, 200, 300 — satisfaz requisitos: $b_{21}, b_{22}, b_{23}, 2a$
resultado esperado: $100 \times 100 + 100 \times 100 = 20.000$

- CT3: 400, 400, 400 — satisfaz requisitos: b31, b32, b33, 1a
resultado esperado: $400 \times 100 = 40.000$
- CT4: 0, 200, 400 — satisfaz requisitos: b11, b22, b33, 0a
resultado esperado: 0
- CT5: 1, 1, 390 — satisfaz requisitos: b21, b22, b23, 3a
resultado esperado: $1 \times 100 + 189 \times 100 + 10 \times 100 = 20.000$

Observações sobre a execução e resultados dos testes (resumo)

Fazer um resumo da execução e comentários e conclusões tiradas do exercício. Só como exemplo, eu posso observar que esses testes não provocam falha quando testado o código janelaC.java, que contém o defeito de assumir que as janelas vem ordenadas $F1 \leq F2 \leq F3$, ou seja, o projeto de testes não está suficientemente bom, pois não está considerando uma característica importante das entradas.

Projeto de testes -

Características consideradas e particionamento para cada característica

1. Consideramos se cada uma das folhas está sobreposta com alguma outra ($|F_i - F_j| < 200$), onde F1, F2 e F3 correspondem as folhas 1, 2 e 3 da janela:

F1 - b1: true
F1 - b2: false

F2 - b1: true
F2 - b2: false

F3 - b1: true
F3 - b2: false

Critério de Combinação

Utilizamos o critério Each Choice.

Requisitos

Os requisitos são:

$$TR = \{ F1-b1, F1-b2, F2-b1, F2-b2, F3-b1, F3-b2 \}$$

Onde F_i-b_j , representam a Folha i (variando entre 1 e 3) e o bloco j (variando entre 1 e 2) para esse particionamento.

Casos de Teste para a satisfação dos requisitos

- CT1: (400, 300, 100), satisfaz os requisitos: F1-b1, F2-b1 e F3-b2.
Resultado esperado: 10000 (true).
- CT2: (0, 300, 400), satisfaz os requisitos: F1-b2, F3-b1 e F2-b1.
Resultado esperado: 30000 (true).
- CT3: (0, 400, 100), satisfaz os requisitos: F2-b2, F1-b1 e F3-b1.
Resultado esperado: 20000 (true).

Obs.: Embora consideramos valores booleanos, nos testes usamos valores inteiros e com isso relacionamos com “true” ou “false”.

Observações sobre a execução e resultados dos testes (resumo)

No caso dos códigos JanelaC.java, Janela2C.java e Janela22.java, os 3 testes apresentaram apenas falhas, sendo que só foram considerados valores válidos.

No caso do código Janela2.java que também utiliza vetores para fazer os cálculos, foram encontradas 2 falhas, todas em casos válidos onde o resultado obtido foi incorreto e 1 erro (em que a terceira folha sobrepõe a primeira em 100), originado por um acesso a uma posição inválida no *array*.

Projeto de testes -

Características consideradas e particionamento para cada característica

1. Consideramos se os valores atribuídos as folhas estavam dentro do esperado, onde F1, F2 e F3 correspondem as folhas 1, 2 e 3. Nessa partição os Fi's assumem valores de 0 (inclusive) até 400 (inclusive), menor que 0 e maior que 400, no qual Fi é a posição da janela no eixo de 0 até 400 (incluindo valores válidos e inválidos).

F1 - b1: $F1 < 0$

F1 - b2: $F1 > 400$

F1 - b3: $F1 \geq 0 \text{ \& } F1 \leq 400$

F2 - b1: $F2 < 0$

F2 - b2: $F2 > 400$

F2 - b3: $F2 \geq 0 \text{ \& } F2 \leq 400$

F3 - b1: $F3 < 0$

F3 - b2: $F3 > 400$

F3 - b3: $F3 \geq 0 \text{ \& } F3 \leq 400$

Critério de Combinação

Utilizamos o método Pairwise.

Requisitos

Os requisitos são:

$$TR = \{ (F1 - b1 \text{ e } F2 - b1), (F1 - b1 \text{ e } F3 - b1), (F1 - b3 \text{ e } F2 - b3), (F1 - b3 \text{ e } F3 - b3), \\ (F1 - b1 \text{ e } F2 - b3), (F1 - b1 \text{ e } F3 - b3), (F1 - b2 \text{ e } F2 - b1), (F1 - b2 \text{ e } F3 - b2), \\ (F1 - b1 \text{ e } F2 - b2), (F1 - b1 \text{ e } F3 - b2), (F1 - b2 \text{ e } F2 - b2), (F1 - b2 \text{ e } F3 - b1), \\ (F1 - b2 \text{ e } F2 - b3), (F1 - b2 \text{ e } F3 - b2), (F1 - b2 \text{ e } F2 - b1), (F1 - b2 \text{ e } F3 - b3), \\ (F1 - b3 \text{ e } F2 - b3), (F1 - b3 \text{ e } F3 - b1), (F1 - b3 \text{ e } F2 - b2), (F1 - b3 \text{ e } F3 - b3), \\ (F1 - b3 \text{ e } F2 - b1), (F1 - b3 \text{ e } F3 - b2) \}$$

Onde Fi-bj, representam a Folha i (variando entre 1 e 3) e bloco j (variando entre 1 e 3) para esse particionamento.

Abaixo, a tabela relativa ao Pairwise:

	F1 b1	F1 b2	F1 b3	F2 b1	F2 b2	F2 b3	F3 b1	F3 b2	F3 b3
F1 b1	----	----	----	----	----	----	----	----	----
F1 b2	----	----	----	----	----	----	----	----	----
F1 b3	----	----	----	----	----	----	----	----	----
F2 b1	x	x	x	----	----	----	----	----	----
F2 b2	x	x	x	----	----	----	----	----	----
F2 b3	x	x	x	----	----	----	----	----	----
F3 b1	x	x	x	x	x	x	----	----	----
F3 b2	x	x	x	x	x	x	----	----	----
F3 b3	x	x	x	x	x	x	----	----	----

Casos de Teste para a satisfação dos requisitos

- CT4: (-1, -1, -1), satisfaz os requisitos: (F1-b1 e F2-b1), (F1-b1 e F3-b1).
Resultado esperado: inválido.
- CT5: (100, 200, 400), satisfaz os requisitos: (F1-b3 e F2-b3), (F1-b3 e F3-b3).
Resultado esperado: 10000.
- CT6: (-1, 300, 400), satisfaz os requisitos: (F1-b1 e F2-b3), (F1-b1 e F3-b3).
Resultado esperado: inválido.
- CT7: (500, -1, 500), satisfaz os requisitos: (F1-b2 e F2-b1), (F1-b2 e F3-b2).
Resultado esperado: inválido.
- CT8: (-1, 500, 500), satisfaz os requisitos: (F1-b1 e F2-b2), (F1-b1 e F3-b2).
Resultado esperado: inválido.
- CT9: (500, 500, -1), satisfaz os requisitos: (F1-b2 e F2-b2), (F1-b2 e F3-b1).
Resultado esperado: inválido.
- CT10: (500, 200, 500), satisfaz os requisitos: (F1-b2 e F2-b3), (F1-b2 e F3-b2).
Resultado esperado: invalido.
- CT11: (500, -1, 300), satisfaz os requisitos: (F1-b2 e F2-b1), (F1-b2 e F3-b3).
Resultado esperado: inválido.
- CT12: (300, 200, -1), satisfaz os requisitos: (F1-b3 e F2-b3), (F1-b3 e F3-b1).
Resultado esperado: inválido.
- CT13: (300, 500, 200), satisfaz os requisitos: (F1-b3 e F2-b2), (F1-b3 e F3-b3).
Resultado esperado: inválido.

- CT14: (100, -1, 500), satisfaz os requisitos: (F1-b3 e F2-b1), (F1-b3 e F3-b2).
Resultado esperado: inválido.

Também achamos importante considerar o caso em que a janela está fechada, para assim verificar como o código responde a esse *input*.

- CT15: (0, 200, 400), caso onde a janela está fechada.
Resultado esperado: 0.

Observações sobre a execução e resultados dos testes (resumo)

No caso do código JanelaC.java, 10 testes apresentaram falhas e 2 obtiveram sucesso, que foram o caso da janela fechada (0,200,400) e o que há uma fresta de 100 no começo (100,200,400). Em Janela2C.java, houve 11 falhas, onde o único sucesso foi no caso de janela fechada.

No caso do código Janela22.java que realiza o cálculo através de *arrays*, nenhum resultado foi satisfatório. Dos 12 casos de teste, houveram 3 falhas originadas pelo cálculo incorreto dos valores (inclusive para os valores válidos). Também foram relatados 9 'erros' onde o programa não conseguiu se recuperar da busca de um valor inválido no vetor. Esse problema poderia ter sido resolvido com uma simples verificação no código para averiguar se o *array* continha a casa que se deseja acessar.

No caso do código Janela2.java que também utiliza vetores para fazer os cálculos, foram encontradas 2 falhas, todas em casos válidos onde o resultado obtido foi incorreto e 10 erros, originados no mesmo problema citado acima para o outro código.

Observação (Anamaria): Quanto ao projeto de teste em si, é importante observar que, se a propriedade "a janela está fechada" é importante, então deve ser uma característica do projeto, com um particionamento correspondente, entrando na estratégia de combinação, etc.

Projeto de testes -

Características consideradas e particionamento para cada característica

As características e particionamentos são descritos de acordo com o quadro abaixo:

Características	True	False
Dupla Sobreposição (pelo menos uma sobreposição entre janelas)	B11	B12
Tripla Sobreposição (pelo menos duas sobreposições entre janelas)	B21	B22
Fechado (área de espaço aberto igual a 0)	B31	B32
Aberto (área de espaço aberto igual a 40000)	B41	B42
Semiaberto (área de espaço aberto entre 0 e 40000)	B51	B52

Critério de Combinação

Pairwise

Requisitos

Abaixo, a tabela do Pairwise:

	B11	B12	B21	B22	B31	B32	B41	B42	B51	B52
B11										
B12										
B21	x	x								
B22	x	x								
B31	x	x	x	x						
B32	x	x	x	x						
B41	x	x	x	x	x	x				
B42	x	x	x	x	x	x				
B51	x	x	x	x	x	x	x	x		
B52	x	x	x	x	x	x	x	x		

Casos de Teste para a satisfação dos requisitos

- CT1: 0, 0, 200 — satisfaz requisitos: (b11,b22), (b11,b32), (b11,b42), (b11,b51), (b22,b32), (b22,b42), (b22,b51), (b32,b42), (b32,b51), (b42, b51)
resultado esperado: 20.000
- CT2: 0, 0, 0 — satisfaz requisitos: (b11,b21), (b11,b41), (b21,b32), (b21,b41), (b22,b31), (b32,b41), (b32,b52), (b41,b52), (b11,b32), (b11,b52), (b21,b52)
resultado esperado: 40.000
- CT3: 0, 50, 100 — satisfaz requisitos: (b21,b51), (b32,b51), (b42,b51), (b21,b32), (b21,b42), (b32,b42)
resultado esperado: 30.000
- CT4: 0, 50, 300 — satisfaz requisitos: (b22,b32), (b22,b51), (b32,b51)
resultado esperado: 15.000
- CT5: 0, 200, 400 — satisfaz requisitos: (b12,b22), (b12,b31), (b12,b42), (b12,b52), (b22,b42), (b22,b52), (b31,b42), (b31,b52), (b42,b52), (b22,b31)
resultado esperado: 0
- Os blocos coloridos na tabela acima são os requisitos insatisfatíveis

Observações sobre a execução e resultados dos testes (resumo)

Todos os testes deram certo apenas na classe JanelaC. Ao executar os cinco testes para a classe Janela22, só o primeiro caso não obteve falha, já no JanelaC2, só o primeiro teve falha. Portanto, a classe Janela2 foi considerada a pior, pois todos falharam.

Projeto de testes -

Características consideradas e particionamento para cada característica

Um particionamento para cada folha (F_i , $i = 1,2,3$):

b1: $F_i = 0$;

b2: $F_i = 200$;

b3: $F_i = 400$;

b4: $F_i > 0$ & $F_i < 200$;

b5: $F_i > 200$ & $F_i < 400$;

Critério de Combinação

Pairwise

Requisitos

b_{ij} onde i é o bloco e j é a folha.

Tabela do Pairwise:

	b1 1	b2 1	b3 1	b4 1	b5 1	b1 2	b2 2	b3 2	b4 2	b52	b13	b2 3	b33	b4 3	b5 3
b11															
b21															
b31															
b41															
b51															
b12	X	X	X	X	X										
b22	X	X	X	X	X										
b32	X	X	X	X	X										
b42	X	X	X	X	X										
b52	X	X	X	X	X										
b13	X	X	X	X	X	X	X	X	X	X					
b23	X	X	X	X	X	X	X	X	X	X					
b33	X	X	X	X	X	X	X	X	X	X					
b43	X	X	X	X	X	X	X	X	X	X					
b53	X	X	X	X	X	X	X	X	X	X					

Nossos requisitos de teste são as combinações dadas pelo PairWise acima (Cada tupla marcada com X).

Casos de Teste para a satisfação dos requisitos

- CT1: (0, 0, 0), satisfaz os requisitos: b11b12, b11b13, b12b13
Resultado esperado:.40.000
- CT2: (200,200,200), satisfaz os requisitos: b21b22, b21b23, b22b23
Resultado esperado:.40.000

- CT3: (400, 400, 400), satisfaz os requisitos: b31b32, b31b33, b32b33
Resultado esperado:.40.000
- CT4: (100, 100, 100), satisfaz os requisitos: b41b42, b41b43, b42b43
Resultado esperado:.40.000
- CT5: (300, 300, 300), satisfaz os requisitos: b51b52, b51b53, b52b53
Resultado esperado:.40.000
- CT6: (0, 200,400), satisfaz os requisitos: b11b22, b11b33, b22b33
Resultado esperado:.0
- CT7: (200, 400, 0), satisfaz os requisitos: b21b32, b21b13, b32b13
Resultado esperado:.0
- CT8: (400, 0, 200), satisfaz os requisitos: b31b12, b31b23, ...(faltou)
Resultado esperado:.0
- CT9: (0, 100, 300), satisfaz os requisitos: b11b42, b11b53, b42b53
Resultado esperado:.10.000
- CT10: (100, 300, 0), satisfaz os requisitos: b41b52, b41b13, b52b13
Resultado esperado:.10.000
- CT11: (300, 0, 100), satisfaz os requisitos: b51b12, b51b43, b12b43
Resultado esperado:.10.000
- CT12: (200, 100, 300), satisfaz os requisitos: b21b42, b21b53, b42b53
Resultado esperado:.20.000
- CT13: (100, 300, 200), satisfaz os requisitos: b41b52, b41b23, b52b23
Resultado esperado:.20.000
- CT14: (300, 200, 100), satisfaz os requisitos: b51b22, b51b43, b22b43
Resultado esperado:.20.000
- CT15: (400, 100, 300), satisfaz os requisitos: b31b42, b31b53, b42b53
Resultado esperado:.10.000
- CT16: (400, 300, 100), satisfaz os requisitos: b31b52, b31b43, b52b43
Resultado esperado:.10.000
- CT17: (300, 400, 400), satisfaz os requisitos: b51b32, b51b33, b32b33
Resultado esperado:.30.000
- CT18: (300, 100, 0), satisfaz os requisitos: b51b42, b51b13, b42b13

Resultado esperado:.30.000

- CT19: (400, 200, 0), satisfaz os requisitos: b31b22, b31b13, b22b13
Resultado esperado:.0
- CT20: (100, 0, 400), satisfaz os requisitos: b41b12, b41b33, b12b33
Resultado esperado:.10.000
- CT21: (100, 200, 300), satisfaz os requisitos: b41b22, b41b53, b22b53
Resultado esperado:.20.000
- CT22: (0, 400, 100), satisfaz os requisitos: b11b32, b11b43, b32b43
Resultado esperado:.10.000
- CT23: (200, 300, 400), satisfaz os requisitos: b21b52, b21b33, b52b33
Resultado esperado:.20.000
- CT24: (0, 300, 200), satisfaz os requisitos: b11b52, b11b23, b52b23
Resultado esperado:.10.000
- CT25: (200, 0, 100), satisfaz os requisitos: b21b12, b21b43, b12b43
Resultado esperado:.20.000
- CT26: (100, 400, 300), satisfaz os requisitos: b41b32, b41b53, b32b53
Resultado esperado:.10.000
- CT27: (300, 400, 200), satisfaz os requisitos: b51b32, b51b23, b32b23
Resultado esperado:.20.000
- CT28: (0, 0, 300), satisfaz os requisitos: b11b12, b11b53, b12b53
Resultado esperado:.20.000
- CT29: (0, 400, 0), satisfaz os requisitos: b11b32, b11b13, b32b13
Resultado esperado:.20.000
- CT30: (200, 100, 200), satisfaz os requisitos: b21b42, b21b23, b42b23
Resultado esperado:.30.000
- CT31: (200, 100, 400), satisfaz os requisitos: b21b42, b21b33, b42b33
Resultado esperado:.10.000

Total: 31 casos de testes.

Observação (Anamaria): Esse conjunto poderia ser menor.

(Anamaria) Tabela de Cobertura dos requisitos Pairwise pelos casos de teste (a ser completada)

	b11	b21	b31	b41	b51	b12	b22	b32	b42	b52	b13	b23	b33	b43	b53
b11															
b21															
b31															
b41															
b51															
b12	1	X	8	X	11										
b22	6	2	X	X	14										
b32	X	7	3	X	X										
b42	9	12	X	4	X										
b52	X	X	X	10, 13	5										
b13	1	7	X	10	X	1	X	7	X	10					
b23	X	2	8	13	X	8	2	X	13	X					
b33	6	X	3	X	X	X	6	3	X	X					
b43	X	X	X	4	11, 14	11	14	X	4	X					
b53	9	12	X	X	5	X	X	X	9, 12	5					

Observações sobre a execução e resultados dos testes (resumo)

Para todos os códigos fornecidos em aula, pelo menos algum teste falhou. Isto mostrou a eficiência do particionamento escolhido junto com o critério de combinação. Abaixo, temos uma tabela resumo da execução de cada teste para cada código. Em adicional, buscamos na web a solução oficial do problema das janelas e executamos todos os testes para este código. No caso da solução oficial, nenhum teste falhou. Os casos de falha, são testes que o código retornou um valor e o esperado era outro. Os casos de erro, são testes que algum erro de ambiente ocorreu, como por exemplo Index Out Of Bounds. Lembrando que o erro também é uma falha, e neste caso, usamos a terminologia do JUnit para diferenciar erro e falha.

Test Case	Janela2	Janela22	JanelaC	JanelaC2	Sol. Oficial
-----------	---------	----------	---------	----------	--------------

1	FALHA	FALHA	SUCESSO	SUCESSO	SUCESSO
2	FALHA	SUCESSO	SUCESSO	SUCESSO	SUCESSO
3	ERRO	FALHA	SUCESSO	SUCESSO	SUCESSO
4	FALHA	FALHA	SUCESSO	SUCESSO	SUCESSO
5	FALHA	FALHA	SUCESSO	SUCESSO	SUCESSO
6	FALHA	FALHA	SUCESSO	SUCESSO	SUCESSO
7	ERRO	FALHA	FALHA	FALHA	SUCESSO
8	FALHA	FALHA	FALHA	FALHA	SUCESSO
9	FALHA	FALHA	SUCESSO	FALHA	SUCESSO
10	FALHA	FALHA	FALHA	FALHA	SUCESSO
11	FALHA	FALHA	FALHA	FALHA	SUCESSO
12	FALHA	FALHA	FALHA	FALHA	SUCESSO
13	FALHA	FALHA	FALHA	FALHA	SUCESSO
14	FALHA	FALHA	FALHA	FALHA	SUCESSO
15	FALHA	FALHA	FALHA	FALHA	SUCESSO
16	FALHA	FALHA	FALHA	FALHA	SUCESSO
17	ERRO	FALHA	SUCESSO	SUCESSO	SUCESSO
18	FALHA	FALHA	FALHA	FALHA	SUCESSO
19	FALHA	FALHA	FALHA	FALHA	SUCESSO
20	FALHA	FALHA	FALHA	FALHA	SUCESSO
21	FALHA	FALHA	SUCESSO	SUCESSO	SUCESSO
22	ERRO	FALHA	FALHA	FALHA	SUCESSO
23	FALHA	FALHA	SUCESSO	SUCESSO	SUCESSO
24	FALHA	FALHA	FALHA	FALHA	SUCESSO
25	FALHA	SUCESSO	FALHA	FALHA	SUCESSO
26	ERRO	FALHA	FALHA	FALHA	SUCESSO
27	ERRO	FALHA	FALHA	FALHA	SUCESSO
28	FALHA	SUCESSO	SUCESSO	SUCESSO	SUCESSO

29	ERRO	FALHA	FALHA	FALHA	SUCESSO
30	SUCESSO	SUCESSO	FALHA	FALHA	SUCESSO
31	FALHA	FALHA	FALHA	FALHA	SUCESSO

- Solução oficial:

http://olimpiada.ic.unicamp.br/passadas/OBI2013/res_fase2_prog/programacao_n2/solucoes/janela1.c

Projeto de testes -

Características consideradas e particionamento para cada característica

1. Para cada parâmetro (F1, F2, F3): os valores válidos vão de 0(inclusive) a 400(exclusive) e os valores inválidos são valores menores que 0 e maiores que 400. Característica para F_i : posição da janela no eixo 'x'
 - a. bloco 1: $F_i < 0$,
 - b. bloco 2: $F_i = 0$
 - c. bloco 3: $0 < F_i < 400$
 - d. bloco 4: $F_i = 400$
 - e. bloco 5: $F_i > 400$

Critério de Combinação

Each Choice

Requisitos

$TRec = \{B11, B21, B31, B41, B51, B12, B22, B32, B42, B52, B13, B23, B33, B43, B53\}$

onde b_{ij} significa bloco i da folha j

Casos de Teste para a satisfação dos requisitos

- CT1: -1, 200, 400 -- satisfaz: B11, B32, B43
 - resultado: inválido
- CT2: 0, 200, 400 -- satisfaz: B21, B32, B43
 - resultado: 0
- CT3: 100, 200, 400 -- satisfaz: B31, B32, B43
 - resultado: 10000
- CT4: 400, 200, 400 -- satisfaz: B41, B32, B43
 - resultado: 20000
- CT5: 500, 400, 400 -- satisfaz: B51, B32, B43
 - resultado = inválido
- CT6: 0, -1, 400 -- satisfaz: B21, B12, B43
 - resultado: inválido
- CT7: 0, 0, 400 -- satisfaz: B21, B22, B43
 - resultado: 20000
- CT8: 0, 400, 400 -- satisfaz: B21, B42, B43

- resultado: 20000
- CT9: 0, 500, 400 -- satisfaz: B21, B52, B43
 - resultado: inválido
- CT10: 0, 200, -1 -- satisfaz: B21, B32, B13
 - resultado: inválido
- CT11: 0, 200, 0 -- satisfaz: B21, B32, B23
 - resultado: 20000
- CT12: 0, 200, 300 -- satisfaz: B21, B32, B33
 - resultado: 10000
- CT13: 0, 200, 500 -- satisfaz: B21, B32, B53
 - resultado: inválido

Observações sobre a execução e resultados dos testes (resumo)

No programa do java2.java não teve nenhum teste com valores válidos que passou. Na execução dos testes ocorreram 6 falhas devido a valores calculados errados e 1 erro java proveniente ao acesso a uma posição inválida no array. Todos os outros 6 testes que passaram foram pelo fato que testavam entradas inválidas e estavam esperando uma exceção.

No programa java22.java ocorreram 8 falhas, nenhum erro java e nos testes que passaram tinha alguns com entradas válidas e inválidas.

No programa janelac2.java ocorreram 10 falhas, nenhum erro java e 3 passaram. Nesse programa os testes com entradas inválidas obtiveram falhas ao invés de erro java, ou seja, o programa aceita valores negativos.

No programa janelaC.java ocorreram 8 falhas. Os testes que falharam eram os casos com as janelas com ordem invertidas ou com entradas inválidas.

Todos os requisitos válidos são testados em casos de testes válidos.

No projeto decidimos optar por 13 casos de testes, mas no each choice bastaria apenas 5 casos de teste para satisfazer todos os requisitos, um exemplo disso está no projeto de teste abaixo (Victor Bauer e Henrique Rappel). A diferença nas escolhas da quantidade de teste foi obtida nos resultados, pelo fato do nosso ter mais casos de testes acabamos verificando um caso de erro quando as janelas estão invertidas.

Observação (Anamaria): O problema em propor 13 casos de teste ao invés de 5 é que não respeita o propósito do Each Choice com esse particionamento. Se o Each Choice não é bom o suficiente, deveria usar Pair Wise ou All Combinations. Ou aperfeiçoar as características consideradas. A ordem é uma característica importante que não consideramos a princípio, mas poderia ser considerada.

Projeto de Testes

-

- Características consideradas e particionamento para cada

característica: Possíveis valores para as respectivas posições de cada folha da janela (J1, J2, J3), considerando valores fora das fronteiras (menores que 0 ou maiores que 400), valores de fronteiras (iguais a 0 ou 400) e valores válidos (entre 1 e 399).

Char.	b1	b2	b3	b4	b5
J1	< 0	= 0	1 até 399	= 400	> 400
J2	< 0	= 0	1 até 399	= 400	> 400
J3	< 0	= 0	1 até 399	= 400	> 400

- Critério de Combinação: Each Choice

- Requisitos de Teste:

- TR_{EC} = Um valor para cada bloco, para cada característica deve ser usado em pelo menos um caso de teste. Chamemos de b_{ij} o caso em que o bloco i é usado com a característica j .
- Temos que $TR_{EC} = \{b_{11}, b_{12}, b_{13}, b_{21}, b_{22}, b_{23}, b_{31}, b_{32}, b_{33}, b_{41}, b_{42}, b_{43}, b_{51}, b_{52}, b_{53}\}$

- Casos de Teste para a satisfação dos requisitos:

1. CT1 = (-1, -1, -1) - Satisfaz requisitos: b_{11}, b_{12}, b_{13}
Resultado esperado: Indefinido

2. CT2 = (0, 0, 0) - Satisfaz requisitos: b21, b22, b23
Resultado esperado: 40.000
3. CT3 = (1, 1, 1) - Satisfaz requisitos: b31, b32, b33
Resultado esperado: 40.000
4. CT4 = (400, 400, 400) - Satisfaz requisitos: b41, b42, b43
Resultado esperado: 40.000
5. CT5 = (401, 401, 401) - Satisfaz requisitos: b51, b52, b53
Resultado esperado: Indefinido

- Observações sobre a execução e resultados dos testes:

- Como somente precisávamos escolher um valor para cada bloco, para cada característica, simplesmente escolhemos um único valor por bloco para todas as características, para facilitar a escolha dos testes.
- Sobre os resultados, em alguns casos, como os testes fora das fronteiras aceitáveis, o teste passou pelo JUnit mas foi avisado que o resultado não era um resultado esperado dentro das fronteiras.
- O teste cumpre os requisitos mas perceba que todas as entradas para todas as características foram iguais. Apesar de de cobrir o Each Choice, realizar testes dessa maneira podem não ser a melhor escolha, pois o programa pode vir a checar casos em que se as entradas são iguais, ele não confere o resto.
- O projeto anterior, também utilizando each choice, foi menos conciso nas escolhas dos casos de testes em relação ao bloco, mas se mostrou mais eficiente. (eficaz é o termo mais adequado)
- **Observações (Anamaria): os testes projetados tem uma característica adicional que é prejudicial à qualidade do conjunto: as posições de todas as folhas são sempre as mesmas em um mesmo teste. Uma implementação que só considerasse um dos valores e o replicasse para os outros 2, não teria esse defeito detectado, por exemplo.**
- **TAREFAS: manter esse conjunto de testes e criar um novo conjunto que satisfaça os requisitos procurando evitar esse problema; explicar o**

tratamento de casos de teste negativos e melhorar os textos marcados em vermelho

- Novos testes -

- Casos de Teste para a satisfação dos requisitos:

6. CT6 = (-1, 0, 1) - Satisfaz requisitos: b11, b22, b33
Resultado esperado: Indefinido
7. CT7 = (0, 1, 400) - Satisfaz requisitos: b21, b32, b43
Resultado esperado: 40.000
8. CT8 = (1, 400, 401) - Satisfaz requisitos: b31, b42, b53
Resultado esperado: 40.000
9. CT9 = (400, 401, -1) - Satisfaz requisitos: b41, b52, b13
Resultado esperado: 40.000
10. CT10 = (401, -1, 0) - Satisfaz requisitos: b51, b12, b23
Resultado esperado: Indefinido

Projeto de testes -

Características consideradas e particionamento para cada característica

Considerando F1, F2 e F3 parâmetros com valores inteiros entre 0 e 400 que determinam a posição de uma determinada janela no eixo, foram decididos os seguintes particionamentos:

- B1: $F_i = 0$
- B2: $F_i > 0 \ \&\& \ < 400$
- B3: $F_i = 400$

Critério de Combinação

Each Choice

Requisitos:

$TR = \{B11, B12, B13, B21, B22, B23, B31, B32, B33\}$, onde B_{ij} representa o bloco i da folha j

Casos de Teste para a satisfação dos requisitos:

- Caso de Teste 1: 0, 0, 0
Satisfaz B11, B12, B13.
Valor Esperado: $400 * 100 = 40000$
- Caso de Teste 2: 100, 250, 320
Satisfaz B21, B22, B23
Valor Esperado: $(100 * 100) + (80 * 100) = 18000$
- Caso de Teste 3: 400, 400, 400
Satisfaz B31, B32, B33
Valor Esperado: $400 * 100 = 40000$

Observações sobre a execução e resultados dos testes (resumo)

Segue o quadro com os valores obtidos na execução dos códigos nos respectivos casos de teste. Os valores possuem fundo verde quando tem o resultado esperado e vermelho quando o contrário. Quando o programa retorna uma exception, ele possui fundo cinza (no único caso em que isso ocorreu, o código tentou acessar um valor em uma posição do buffer maior do que o tamanho do próprio).

	CASO DE TESTE 1	CASO DE TESTE 2	CASO DE TESTE 3
Janela2.java	20000	42000	<i>Retornou exception</i>
Janela22.java	20000	45000	60000
JanelaC.java	40000	18000	40000
JanelaC2.java	40000	18000	40000

Não podemos ter certeza sobre a validade dos códigos da JanelaC e JanelaC2, especialmente com a pequena quantidade de casos de teste utilizados, mas podemos verificar que Janela2.java e Janela22.java não obtiveram os resultados esperados em nenhum caso de teste.

Projeto de testes -

Características consideradas e particionamento para cada característica

P1

Valores válidos para as folhas: seus valores (inteiros) vão de 0 (inclusive) a 400 (inclusive). Característica para F_i : posição da folha no eixo 0..400 (valores válidos e inválidos).

1. bloco1: Valores Válidos para as folhas
 - a. $0 \leq F_i \leq 400$
2. bloco2: Valores inválidos para as folhas
Característica para F_i : posição da folha fora do eixo 0..400
 - a. $F_i < 0$ ou $F_i > 400$

P2

Valores para a área em relação aos valores das folhas.

3. bloco3: Área nula.
4. bloco4: Área máxima.
5. bloco5: Área intermediária.

Critério de Combinação

Each Choice

Requisitos

$TREC = \{b_{11}, b_{12}, b_{13}, b_{21}, b_{22}, b_{23}, b_3, b_4, b_5\}$, onde b_{ij} significa bloco i da folha j

Casos de Teste para a satisfação dos requisitos

- CT1: 401, -1, 410 — satisfaz requisitos: b_{21} , b_{22} , b_{23}
É um teste negativo, não tem como prever o que pode retornar.
Resultado esperado: Inválido
- CT2: 50, 210, 390 — satisfaz requisitos: b_{11} , b_{12} , b_{13} , b_5
Resultado esperado: $A = 6.000$

- CT3: 70, 70, 70 — satisfaz requisitos: b11, b12, b13, b4
Resultado esperado: A = 40.000
- CT4: 0, 0, 200 — satisfaz requisitos: b11, b12, b13, b5
Resultado esperado: A = 20.000
- CT5: 0, 200, 400 — satisfaz requisitos: b11, b12, b13, b3
Resultado esperado: A = 0

Observações sobre a execução e resultados dos testes (resumo)

Podemos verificar que foram desenvolvidos casos de testes que satisfaziam cada um dos blocos, já que usando o Each Choice, cada bloco é considerado um requisito. Ainda assim, os testes não foram escolhidos de forma eficientes, apesar dos testes mostrarem que os códigos JanelaC2, Janela2, Janela22 possuem erros, o código Janela C obteve sucesso em todos os testes mesmo não estando correto. Conforme confirmado durante a aula, o código janelaC.java, contém o defeito de assumir que as janelas vem ordenadas $F1 \leq F2 \leq F3$.

- JanelaCTest: Passou em todos os testes (CT1, CT2, CT3, CT4, CT5). Inclusive o caso de teste negativo, o CT1, demonstrando que aceita valores inválidos.
- JanelaC2Test: Não passou apenas no teste 4 (CT4). Retornando um valor errado para a área calculada.
- Janela2Test: Falhou em todos os testes (CT1, CT2, CT3, CT4, CT5).
- Janela22Test: Falhou em todos os testes (CT1, CT2, CT3, CT5) exceto o CT4 que testa valores intermediários para a área.

Observação (Anamaria): CT4 está cobrindo exatamente os mesmos requisitos que CT2. É considerado um caso de teste redundante de acordo com os particionamentos e critério de combinação escolhido. Isso não deveria acontecer.

Projeto de testes -

Características consideradas e particionamento para cada característica

1. Para cada parâmetro (F1, F2, F3): seus valores (inteiros) vão de 0 (inclusive) a 400 (inclusive). Característica para Fi : posição da janela no eixo 0..400 (só incluindo valores válidos)

Q1 : B01 → posição da folha 1 = 0
B11 → posição da folha 1 ≥ 0 e posição da folha 1 < 200
B21 → posição da folha 1 ≥ 200 e posição da folha 1 < 400
B31 → posição da folha 1 = 400

Q2 : B02 → posição da folha 2 = 0
B12 → posição da folha 2 ≥ 0 e posição da folha 2 < 200
B22 → posição da folha 2 ≥ 200 e posição da folha 2 < 400
B32 → posição da folha 2 = 400

Q3 : B03 → posição da folha 3 = 0
B13 → posição da folha 3 ≥ 0 e posição da folha 3 < 200
B23 → posição da folha 3 ≥ 200 e posição da folha 3 < 400
B33 → posição da folha 3 = 400

Critério de Combinação

All Combination

Requisitos

TR = {(b01b02b03), (b01b02b13), (b01b02b23), (b01b02b33), (b01b12b03), (b01b12b13), (b01b12b23), (b01b12b33), (b01b22b03), (b01b22b13), (b01b22b23), (b01b22b33), (b01b32b03), (b01b32b13), (b01b32b23), (b01b32b33), (b11b02b03), (b11b02b13), (b11b02b23), (b11b02b33), (b11b12b03), (b11b12b13), (b11b12b23), (b11b12b33), (b11b22b03), (b11b22b13), (b11b22b23), (b11b22b33), (b11b32b03), (b11b32b13), (b11b32b23), (b11b32b33), (b21b02b03), (b21b02b13), (b21b02b23), (b21b02b33), (b21b12b03), (b21b12b13), (b21b12b23), (b21b12b33), (b21b22b03), (b21b22b13), (b21b22b23), (b21b22b33), (b21b32b03), (b21b32b13), (b21b32b23), (b21b32b33), (b31b02b03), (b31b02b13), (b31b02b23), (b31b02b33), (b31b12b03), (b31b12b13),

(b31b12b23), (b31b12b33), (b31b22b03), (b31b22b13), (b31b22b23), (b31b22b33), (b31b32b03), (b31b32b13), (b31b32b23), (b31b32b33)}.

Casos de Teste para a satisfação dos requisitos

- CT1: 0, 0, 0 — satisfaz requisitos:b01b02b03
Resultado esperado: 40.000
- CT2: 0, 0, 100 — satisfaz requisitos:b01b02b13
Resultado esperado: 30.000
- CT3: 0, 0, 200 — satisfaz requisitos:b01b02b23
Resultado esperado: 20.000
- CT4: 0, 0, 400 — satisfaz requisitos:b01b02b33
Resultado esperado: 20.000
- CT5: 0, 100, 0 — satisfaz requisitos:b01b12b03
Resultado esperado: 30.000
- CT6: 0, 100, 100 — satisfaz requisitos:b01b12b13
Resultado esperado: 30.000
- CT7: 0, 100, 200 — satisfaz requisitos:b01b12b23
Resultado esperado: 20.000
- CT8: 0, 100, 400 — satisfaz requisitos:b01b12b33
Resultado esperado: 10.000
- CT9: 0, 200, 0 — satisfaz requisitos:b01b22b03
Resultado esperado: 20.000
- CT10: 0, 200, 100 — satisfaz requisitos:b01b22b13
Resultado esperado: 20.000
- CT11: 0, 200, 200 — satisfaz requisitos:b01b22b23
Resultado esperado: 20.000
- CT12: 0, 200, 400 — satisfaz requisitos:b01b22b33
Resultado esperado: 0

- CT13: 0, 400, 0 — satisfaz requisitos:b01b32b03
Resultado esperado: 20.000
- CT14: 0, 400, 100 — satisfaz requisitos:b01b32b13
Resultado esperado: 10.000
- CT15: 0, 400, 200 — satisfaz requisitos:b01b32b23
Resultado esperado: 0
- CT16: 0, 400, 400 — satisfaz requisitos:b01b32b33
Resultado esperado: 20.000
- CT17: 100, 0, 0 — satisfaz requisitos:b11b02b03
Resultado esperado: 30.000
- CT18: 100, 0, 100 — satisfaz requisitos:b11b02b13
Resultado esperado: 30.000
- CT19: 100, 0, 200 — satisfaz requisitos:b11b02b23
Resultado esperado: 20.000
- CT20: 100, 0, 400 — satisfaz requisitos:b11b02b33
Resultado esperado: 10.000
- CT21: 100, 100, 0 — satisfaz requisitos:b11b12b03
Resultado esperado: 30.000
- CT22: 100, 100, 100 — satisfaz requisitos:b11b12b13
Resultado esperado: 40.000
- CT23: 100, 100, 200 — satisfaz requisitos:b11b12b23
Resultado esperado: 30.000
- CT24: 100, 100, 400 — satisfaz requisitos:b11b12b33
Resultado esperado: 20.000
- CT25: 100, 200, 0 — satisfaz requisitos:b11b02b03
Resultado esperado: 20.000
- CT26: 100, 200, 100 — satisfaz requisitos:b11b02b13
Resultado esperado: 30.000

- CT27: 100, 200, 200 — satisfaz requisitos:b11b02b23
Resultado esperado: 30.000
- CT28: 100, 200, 400 — satisfaz requisitos:b11b02b33
Resultado esperado: 10.000
- CT29: 100, 400, 0 — satisfaz requisitos:b11b32b03
Resultado esperado: 20.000
- CT30: 100, 400, 100 — satisfaz requisitos:b11b32b13
Resultado esperado: 20.000
- CT31: 100, 400, 200 — satisfaz requisitos:b11b32b23
Resultado esperado: 10.000
- CT32: 100, 400, 400 — satisfaz requisitos:b11b32b33
Resultado esperado: 20.000
- CT33: 200, 0, 0 — satisfaz requisitos:b21b02b03
Resultado esperado: 20.000
- CT34: 200, 0, 100 — satisfaz requisitos:b21b02b13
Resultado esperado: 20.000
- CT35: 200, 0, 200 — satisfaz requisitos:b21b02b23
Resultado esperado: 20.000
- CT36: 200, 0, 400 — satisfaz requisitos:b21b02b33
Resultado esperado: 0
- CT37: 200, 100, 0 — satisfaz requisitos:b21b12b03
Resultado esperado: 20.000
- CT38: 200, 100, 100 — satisfaz requisitos:b21b12b13
Resultado esperado: 30.000
- CT39: 200, 100, 200 — satisfaz requisitos:b21b12b23
Resultado esperado: 30.000
- CT40: 200, 100, 400 — satisfaz requisitos:b21b12b33
Resultado esperado: 10.000

- CT41: 200, 200, 0 — satisfaz requisitos:b21b02b03
Resultado esperado: 20.000
- CT42: 200, 200, 100 — satisfaz requisitos:b21b02b13
Resultado esperado: 30.000
- CT43: 200, 200, 200 — satisfaz requisitos:b21b02b23
Resultado esperado: 40.000
- CT44: 200, 200, 400 — satisfaz requisitos:b21b02b33
Resultado esperado: 20.000
- CT45: 200, 400, 0 — satisfaz requisitos:b21b32b03
Resultado esperado: 0
- CT46: 200, 400, 100 — satisfaz requisitos:b21b32b13
Resultado esperado: 10.000
- CT47: 200, 400, 200 — satisfaz requisitos:b21b32b23
Resultado esperado: 20.000
- CT48: 200, 400, 400 — satisfaz requisitos:b21b32b33
Resultado esperado: 20.000
- CT49: 400, 0, 0 — satisfaz requisitos:b31b02b03
Resultado esperado: 20.000
- CT50: 400, 0, 100 — satisfaz requisitos:b31b02b13
Resultado esperado: 20.000
- CT51: 400, 0, 200 — satisfaz requisitos:b31b02b23
Resultado esperado: 0
- CT52: 400, 0, 400 — satisfaz requisitos:b31b02b33
Resultado esperado: 20.000
- CT53: 400, 100, 0 — satisfaz requisitos:b31b12b03
Resultado esperado: 10.000
- CT54: 400, 100, 100 — satisfaz requisitos:b31b12b13
Resultado esperado: 20.000

- CT55: 400, 100, 200 — satisfaz requisitos:b31b12b23
Resultado esperado: 10.000
- CT56: 400, 100, 400 — satisfaz requisitos:b31b12b33
Resultado esperado: 20.000
- CT57: 400, 200, 0 — satisfaz requisitos:b31b02b03
Resultado esperado: 20.000
- CT58: 400, 200, 100 — satisfaz requisitos:b31b02b13
Resultado esperado: 10.000
- CT59: 400, 200, 200 — satisfaz requisitos:b31b02b23
Resultado esperado: 20.000
- CT60: 400, 200, 400 — satisfaz requisitos:b31b02b33
Resultado esperado: 20.000
- CT61: 400, 400, 0 — satisfaz requisitos:b31b32b03
Resultado esperado: 20.000
- CT62: 400, 400, 100 — satisfaz requisitos:b31b32b13
Resultado esperado: 20.000
- CT63: 400, 400, 200 — satisfaz requisitos:b31b32b23
Resultado esperado: 20.000
- CT64: 400, 400, 400 — satisfaz requisitos:b31b32b33
Resultado esperado: 40.000

Observações sobre a execução e resultados dos testes (resumo)

- O critério utilizado apesar de completo foi muito trabalhoso, uma ferramenta foi usada para escrever os códigos de todos os casos de teste, por este motivo a dupla considerou a escolha do critério de todas as combinação ruim.
- Os quatro códigos testados apresentaram muitas falhas.
 - Janela2.java → 16 erros e 42 falhas
 - Janela22.java → 0 erros e 49 falhas
 - JanelaC.java → 0 erros e 45 falhas
 - JanelaC2.java → 0 erros e 50 falhas

- Também foi notado que, uma vez o teste codificado, não há necessidade de o fazer novamente. Ou seja, apesar de ser trabalhoso escrever o código, esta técnica é melhor que testar “na mão”.
- Observação (Anamaria): Por outro lado, se os defeitos dos códigos fossem mais sutis, esse conjunto de testes teria uma boa probabilidade de exercitá-los, ao passo que outros projetos menos rigorosos possivelmente, não.