

Part 1

3. Answer the following questions for the method `search()` below:

```
public static int search (List list, Object element)
// Effects: if list or element is null throw NullPointerException
// else if element is in the list, return an index
// of element in the list; else return -1
// for example, search ([3,3,1], 3) = either 0 or 1
// search ([1,7,5], 2) = -1
```

Base your answer on the following characteristic partitioning:

Characteristic: Location of element in list

- Block 1: element is first entry in list
- Block 2: element is last entry in list
- Block 3: element is in some position other than first or last

a) "Location of element in list" fails the disjointness property. Give an example that illustrates this.

اگر لیست طول ۱ داشته باشد و المان مورد نظر همان تک عنصر داخل لیست باشد، بلاک‌های ۱ و ۲ با هم برقرار هستند. مانند

List = {1} element = 1

b) "Location of element in list" fails the completeness property. Give an example that illustrates this.

بلاکی برای عدم وجود المان جست‌وجو در لیست در نظر گرفته نشده است، مانند

List = {1,2,3} element = 5

هیچ کدام از بلاک‌ها شامل این حالت نمی‌شوند

c) Supply one or more new partitions that capture the intent of "Location of element in list" but do not suffer from completeness or disjointness problems.

Characteristic: element is first

Block 1: True

Block 2: False

Characteristic: element is last

Block 1: True

Block 2: False

Characteristic: element is in list, but it's not first nor last

Block 1: True

Block 2: False

Characteristic: element is not in list

Block 1: True

Block 2: False

4. Derive input space partitioning test inputs for the `GenericStack` class assuming the following method signatures:

- `public GenericStack ();`
- `public void push (Object X);`
- `public Object pop ();`
- `public boolean isEmpty ();`

Assume the usual semantics for the `GenericStack`. Try to keep your partitioning simple and choose a small number of partitions and blocks.

Explicit: Object x

Implicit (state): size, list of elements

Object x

Characteristic: is x null

Block 1: True

x = null

Block 2: False

x = 31

Size

Characteristic: what is the size of stack

Block 1: 0

List = {}

Block 2: 1

List = {56} Size = 1

Block 3: more than 1

List = {52,3,2,64} Size = 4

Characteristic: is stack empty

Block 1: True

List = {} Size = 0

Block 2: False

List = {2,45} Size = 2

List of elements

Characteristic: does the list contain null elements

Block 1: True

List = {2,4,23,null,53,null}

Block 2: False

List = {2,5,64}

Part 2

4. Answer the following questions for the method intersection() below:

```
public Set intersection (Set s1, Set s2)
// Effects:  If s1 or s2 is null throw NullPointerException
//   else return a (non null) Set equal to the intersection
//   of Sets s1 and s2
```

Characteristic: Validity of s1

- s1 = null
- s1 = {}
- s1 has at least one element

Characteristic: Relation between s1 and s2

- s1 and s2 represent the same set
- s1 is a subset of s2
- s2 is a subset of s1
- s1 and s2 do not have any elements in common

a) Does the partition “Validity of s1” satisfy the completeness property? If not, give a value for s1 that does not fit in any block.

It is complete

b) Does the partition “Validity of s1” satisfy the disjointness property? If not, give a value for s1 that fits in more than one block.

It is disjoint

c) Does the partition “Relation between s1 and s2” satisfy the completeness property? If not, give a pair of values for s1 and s2 that does not fit in any block.

It is complete

d) Does the partition “Relation between s1 and s2” satisfy the disjointness property? If not, give a pair of values for s1 and s2 that fits in more than one block.

It is NOT disjoint

S1 = {} S2 = {}

In this example, s1 and s2 both represent the same set (b1) AND they have no element in common (b4)

e) If the “Base Choice” criterion were applied to the two partitions (exactly as written), how many test requirements would result?

$$1 + (3-1) + (4-1) = 1+2+3 = 6$$

f) Revise the characteristics to eliminate any problems you found.

Characteristic: relation between s1 and s2

- s1 and s2 represent the same non-empty set
- s1 is a non-empty subset of s2
- s2 is a non-empty subset of s1
- s1 and s2 do not have any elements in common (one or both could be empty)

5. Use the following characteristics and blocks for the questions below.

Characteristics	Block 1	Block 2	Block 3	Block 4
Value 1	< 0	0	> 0	
Value 2	< 0	0	> 0	
Operation	+	−	×	÷

a) Give tests to satisfy the Each Choice criterion.

Value1 = -1	Value2 = -2	Operation = +	(1,1,1)
Value1 = 0	Value2 = 0	Operation = -	(2,2,2)
Value1 = 2	Value2 = 3	Operation = *	(3,3,3)
Value1 = -5	Value2 = 6	Operation = /	(1,3,4)

b) Give tests to satisfy the Base Choice criterion. Assume base choices are Value 1: > 0, Value 2: > 0, and Operation: +.

Value1 = 3	Value2 = 2	Operation = +	(base test)
Value1 = 3	Value2 = 0	Operation = +	(Value2 = 0)
Value1 = 3	Value2 = -1	Operation = +	(Value2 < 0)
Value1 = 0	Value2 = 2	Operation = +	(Value1 = 0)

Value1 = -3	Value2 = 2	Operation = +	(Value1 < 0)
Value1 = 3	Value2 = 2	Operation = -	(operation -)
Value1 = 3	Value2 = 2	Operation = *	(operation *)
Value1 = 3	Value2 = 2	Operation = /	(operation /)

c) How many tests are needed to satisfy the All-Combinations criterion? (Do not list all the tests.)

$$3 * 3 * 4 = 36$$

d) Give tests to satisfy the Pair-Wise Coverage criterion.

Answers will be (Value1, Value2, Operation) total of 12 tests

(-1, -2, +)	(-1, 0, -)	(-1, 2, *)	(-1, -2, /)
(0, 0, +)	(0, 2, -)	(0, -2, *)	(0, 0, /)
(1, 2, +)	(1, -2, -)	(1, 0, *)	(1, 2, /)

Part 3

$$\text{Total tests} = 4 * 4 = 16$$

(A1, B1, C1, D1)	(A1, B2, C2, D2)	(A1, B3, C3, D3)	(A1, B4, C4, D4)
(A2, B1, C2, D3)	(A2, B2, C3, D4)	(A2, B3, C4, D1)	(A2, B4, C1, D2)
(A3, B1, C3, D1)	(A3, B2, C4, D2)	(A3, B3, C1, D3)	(A3, B4, C2, D4)
(A4, B1, C4, D3)	(A4, B2, C1, D4)	(A4, B3, C2, D1)	(A4, B4, C3, D2)

Part 4

Derive input space partitioning test inputs for the BoundedQueue class with the following method signatures:

```
public BoundedQueue (int capacity); // The maximum number of elements
```

Assume the usual semantics for a queue with a fixed, maximal capacity. Try to keep your partitioning simple - choose a small number of partitions and blocks.

Capacity

Value of capacity

Block 1: 0 capacity = 0

Block 2: 1 capacity = 1

Block 2: > 1 capacity = 4 base

Size (implicit in class)

Is queue empty

Block 1: True Capacity = 10 Size = 0

Block 2: False Capacity = 10 Size = 2 base

Is queue full

Block 1: True Capacity = 10 Size = 10

Block 2: False Capacity = 10 Size = 4 base