Homework 2: Input Space Partitioning for BoundedQueue

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

- __init__(self, capacity) # The Maximum number of elements
- enqueue(self, o)
- dequeue(self)
- is_empty(self)
- is_full(self)

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.

(a) List all of the input variables, including the (abstract) state variables.

(b) Define the characteristics of the input variables. Make sure you cover all input variables

(c) Partition the characteristics into blocks. Designate one block in each partition as the "Base" block.

(d) Define values for each block.

(e) Define a test set that satisfies Base Choice Coverage (BCC). Write your tests with the values from the previous step. Be sure to include the test oracles.

(a) List all of the input variables, including the (abstract) state variables.

input variable：

- o：enqueue 方法的輸入變數，在 enqueue() 方法中用來表示要加入佇列的元素。
- cap：BoundedQueue 類的建構子參數，在 __init__() 方法中作為參數傳入的佇列容量大小。

(abstract) state variable：

- capacity：佇列的容量大小，即佇列最多可以容納的元素個數。
- elements：一個長度為 capacity 的列表，用來存儲佇列中的元素。
- size：當前佇列中的元素個數。
- front：佇列的前端索引，指向佇列中第一個元素。
- back：佇列的後端索引，指向佇列中最後一個元素的下一個位置。

(b) Define the characteristics of the input variables. Make sure you cover all inpu t variables.

Method	Params	Returns	Values	Exception	Ch ID	Characteristic
BoundedQue ue	int				C1	Constructor
				ValueError	C2	Arg(cap) < 0
enque	object				C3	Add object o to the queue
				RuntimeError	C7	If the queus is full (size == capacity)
				TypeError	C4	Object o is null
dequeue	state	Object	object/ null		C5	Remove and return the oldest object from queue
				RuntimeError	C6	If queue is empty (size == 0)
is_empty	state	Boolean	true/ false		C6	If queue is empty
is_full	state	Boolean	true/ false		C7	If queue is full

(c) Partition the characteristics into blocks. Designate one block in each partition as the "Base" block.

ID	Characteristic	BoundedQueue(int)	enque(Object)	deque()	is_empty()	is_full()	Partition
C1	Constructor	X	X	X	X	X	
C2	Arg(cap) < 0	X	X	X	X	X	{true,false}
C3	Add object o to the queue		X				{true,false}
C4	Object o is null		X				{true,false}
C5	Remove the oldest object from queue			X			{true,false}
C6	If queue is empty			X	X		{true,false}
C7	If queue is full		X			X	{true,false}
	Base Block	create a new queue, cap >0	not full, enque a not null object	not empty, dequeue	not empty	not null	

(d) Define values for each block.

ID	Characteristic	BoundedQueue(int)	enque(Object)	deque()	is_empty()	is_full()	Partition
C1	Constructor	X	X	X	X	X	
C2	Arg(cap) < 0	X	X	X	X	X	{true,false}
C3	Add object o to the queue		X				{true,false}
C4	Object o is null		X				{true,false}
C5	Remove the oldest object from queue			X			{true,false}
C6	If queue is empty			X	X		{true,false}
C7	If queue is full		X			X	{true,false}
	Base Block	{F} create a new queue, cap >0	{FTFF} not full, enque a not null object	{FTF} not empty, dequeue	{FF} not empty	{FF} not null	

(e) Define a test set that satisfies Base Choice Coverage (BCC). Write your tests w ith the values from the previous step. Be sure to include the test oracles.

Method	Characteristics	Test Requirements	Infeasible TRs	Revised TRs	# TRs
BoundedQueue(int)	C1 C2	{**F**, T}			2
enque(Object)	C1 C2 C3 C4 C7	{**FTFF**, FTFT, FTTF, FTTT}	TTFF, TTFT, TTTF, TTTT	TTFF->FTFF TTFT->FTFT TTTF->FTTF TTTT->FTTT	4
deque()	C1 C2 C5 C6	{**FTF**, FTT}	TTT, TTF, TFT, TFF	TTT->FTT TTF->FTF TFT->FFT TFF-> FFF	2
is_empty()	C1 C2 C6	{**FF**, FT}	TT, TF	TT->FT TF->FF	2
is_full()	C1 C2 C7	{**FF**, FT}	TT, TF	TT->FT TF->FF	2

Test BoundedQueue C1, C2

- BoundedQueue Base : F(arg >= 0)
  # no exception
- BoundedQueue2 : T(arg < 0)
  #raise ValueError

Test Enqueue C1, C2, C3, C4, C7

- enqueue base: FTFF (when queue is not full, enqueue an object)
  # no exception
- enqueue 2. FTFT (queue is full, enqueue an object)
  # raise RuntimeError
- enqueue 3. FTTF (queue is not full, enqueue null object)
  # raise TypeError
- enqueue 4. FTTT (queue is full, enqueue null object)
  #raise TypeError

Test Dequeue C1, C2, C5, C6

- base: FTF (queue is not empty, dequeue)
  #no exception
- dequeue 2: FTT (queue is empty, dequeue)
  # raise RuntimeError

Test isEmpty C1, C6, C7

- base FF (new a not empty queue)
  #return False
- 2. FT (new an empty queue)
  #return True

Test isFull C1, C2, C7

- base FF (new a not full queue)
  #return False
- 2. FT (new a full queue)
  #return True