

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.

Input variables	<code>cap(int)</code> , <code>o(unkown)</code>
state variables	<code>self.capacity (int)</code> <code>self.elements (list of objects)</code> <code>self.size (int)</code> <code>self.front (int)</code> <code>self.back (int)</code>

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

Function	Params	Returns	Values	Exception	Ch I D	Characteristic	Covered by
BoundedQueue	int				C1	If input cap >= 0, set the capacity	
				ValueError		Input cap < 0	C1
enqueue	o(object)				C2	Push new o(object) only if queue is not full and o(object) is not None	
				TypeError	C3	Check o(object) whether o(object) is None or not None	
				RuntimeError	C4	Queue is not full	
dequeue	state	object	object		C5	Pop and Return oldest object in the queue if queue is not empty	
				RuntimeError	C6	Queue is not empty	
is_empty	state	boolean	true, false		C7	Return true if queue is empty	
is_full	state	boolean	true, false		C8	Return true if queue is full	

註1: 因為在code中，enqueue()判斷queue是否full並沒有呼叫is_full()，因此將他視為獨立的Characteristic，dequeue()也是同理，dequeue()判斷queue是否empty並沒有呼叫is_empty()。

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

Function	Ch I D	Characteristic	block1(Base)	block2	block3
BoundedQueue	C1	If input cap >= 0, set the capacity	cap > 0	cap < 0	cap = 0
enqueue	C2	Push new o(object) only if queue is not full and o(object) is not None	True: Successfully push new o(object)	False: Fail to push new o(object)	
	C3	Check o(object) whether o(object) is None or not None	True: o(object) is not None	False: o(object) is None	
	C4	Queue is not full	True: Queue is not full	False: Queue is full	
dequeue	C5	Pop and Return oldest object in the queue if queue is not empty	True: Successfully pop and return oldest o(object)	False: Fail to pop and return oldest o(object)	
	C6	Queue is not empty	True: Queue is not empty	False: Queue is empty	
is_empty	C7	Queue is not empty	True: Queue is empty	False: Queue is not empty	
is_full	C8	queue is full	True: Queue is full	False: Queue is not full	

(d) Define values for each block.

Function	Ch I D	Characteristic	block1(Base)	block2	block3
BoundedQueue	C1	If input cap >= 0, set the capacity	5 is_empty = true is_full = false	-5 ValueError	(註2) 0 is_empty = True is_full = True
enqueue	C2	Push new o(object) only if queue is not full and o(object) is not None	True: Successfully push new o(object)	False: Fail to push new o(object)	
	C3	Check o(object) whether o(object) is None or not None	True: o(object) is not None	False: o(object) is None TypeError	
	C4	Queue is not full	True: Queue is not full	False: Queue is full RuntimeError	
dequeue	C5	Pop and Return oldest object in the queue if queue is not empty	True: Successfully pop and return oldest o(object)	False: Fail to pop and return oldest o(object)	
	C6	Queue is not empty	True: Queue is not empty	False: Queue is empty RuntimeError	
is_empty	C7	Queue is not empty	True: Queue is empty	False: Queue is not empty	
is_full	C8	queue is full	True: Queue is full	False: Queue is not full	

註2: 按照助教提供的BoundedQueue.py，當cap = 0時，is_empty = True，is_full = True

(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.

Function	Ch ID	Base	Test Requirements	Infeasible TRs	#TRs
BoundedQueue	C1	{5}	{5}, {-5}, {0}		3
enqueue	C2, C3, C4	{T, T, T}	{T, T, T}, {F, F, T}, {F, T, F}	{F, T, T}	3
dequeue	C5, C6	{T, T}	{T, T}, {F, F}	{T, F}, {F, T}	2
is_empty	C7	{T}	{T}, {F}		2
is_full	C8	{F}	{F}, {T}		2