|  |  |
| --- | --- |
| **void** inject(QueueOfT& q, T& x);//! q's parameter mode \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_//! x's parameter mode \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_//! requires: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_//! ensures: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_//! decreasing: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | **E3P1 – Fill in reasoning table for a recursive operation (Points: 30 – 2 points/each)** **To Do:**   1. Fill in the operation’s *External Contract* (to the left) 2. Fill in *Assume* Cells: A2, A4, A5, A6, A7, A8 3. Fill in *Confirm* cells: C4, C5,C6, C8   Note: Cells A0, A1, A3, C0, C1, and C3 have been done for you |

|  |  |  |  |
| --- | --- | --- | --- |
| **State** | **Code** | **Assume** | **Confirm** |
| 0 |  | *A0*:  true *A0.1* | *C0*:  true *C0.1* |
|  | **if**(q.length() == 0) { | *For Gradescope auto grading, leave only 1 space around operators* |  |
| 1 |  | *A1*:  (|q0| = 0) -> (q1 = q0 ^ x1 = x0) *A1.1* | *C1*:  (|q0| = 0) -> true *C1.1* |
|  | q.enqueue(x); |  |  |
| 2 |  | *A2*: | *C2*:  *leave blank* |
|  | } **else** { |  |  |
| 3 |  | *A3*:  (|q0| > 0) -> (q3 = q0 ^ x3 = x0) *A3.1* | *C3*:  (|q0| > 0) -> true *C3.1* |
|  | T y; |  |  |
| 4 |  | *A4*: | *C4*: |
|  | q.dequeue(y); |  |  |
| 5 |  | *A5*: | *C5*: |
|  | inject(q, x); |  |  |
| 6 |  | *A6*: | *C6*: |
|  | q.enqueue(y); |  |  |
| 7 |  | *A7*: | *C7*: |
|  | } // end if |  |  |
| 8 |  | *A8*: | *C8*: |