Started on	Wednesday, 3 May 2023, 21:51
State	Finished
Completed on	Wednesday, 3 May 2023, 22:07
Time taken	16 mins 4 secs
Grade	Not yet graded

Information

Consider the following piece of code for the question below:

```
class Y<Key extends Comparable<Key>> {
    private Key[] A = (Key[]) new Comparable[1];
    private int lo, hi, N;
    public void insert(Key in)
    {
        A[hi] = in;
        hi = hi + 1;
        if (hi == A.length) hi = 0;
        N = N + 1;
        if (N == A.length) rebuild();
    }
    public Key remove() // assumes Y is not empty
    {
        Key out = A[lo];
        A[lo] = null;
        lo = lo + 1;
        if (lo == A.length) lo = 0;
        N = N - 1;
        return out;
    }
    private void rebuild()
    {
        // The line below is essentially:
        // Key[] tmp = new Key[2*A.length]
        // with keys being comparable.
        Key[] tmp = (Key[]) new Comparable[2 * A.length];
        for (int i = 0; i < N; i++)
            tmp[i] = A[(i + lo) % A.length];
        A = tmp;
        lo = 0;
        hi = N;
    }
}
```

Question 1

Correct

Mark 1.00 out of 1.00

Class Y behaves like which well-known data type?

- a. Priority queue
- b. Queue ✓
- c. Union–Find
- d. Stack

Question 2

Correct

Mark 1.00 out of 1.00

Write the body of a method `int size()` that returns the number of elements in the data structure.

- a. `return A.length;`
- b. `return hi - lo;`
- c. `return A[N];`
- d. `return N;` ✓

Question 3

Correct

Mark 1.00 out of 1.00

Which invariant does the data structure maintain after every `remove` and `insert`? (Recall that an invariant is a condition that the data structure ensures is true after each operation.)

- a. `hi < N`
- b. `N < A.length` ✓
- c. `hi == N`
- d. `lo < hi`

Question **4**

Complete

Marked out of 1.00

Assume that the contents of `A` are

```
    +---+---+---+---+
A == | 3 | 8 | 4 | 1 |
    +---+---+---+---+
```

and `lo == 3`, `hi ==2`, `N == 2`.

Is the above situation something that can arise from sequence of `insert` and `remove`? If yes, give such a sequence. Otherwise, explain why not.

```
No, N indicates that there are only 2 values in A even though it looks otherwise.
```

Question **5**

Complete

Marked out of 1.00

In the situation from the previous question, what are the contents of `A`, `lo`, and `hi` after executing `rebuild` in the given state?

```
A = [3, 0, 4, 1, null, null, null, null]
lo = 0
hi = 4
```

Question 6

Complete

Marked out of 1.00

Draw the data structure (including the contents of `A` and the values of `hi`, `lo`, and `N`) after the following operations, and indicate how many times `rebuild` were called:

```
Y y= new Y(); // in Python: y = Y()
y.insert(1);
y.remove();
y.insert(2);
y.remove();
y.insert(3);
```

```
A = [3, null]
lo = 0
hi = 1
N = 1
```

Question 7

Correct

Mark 1.00 out of 1.00

Given the following partially known sequence of operations, what are the possible contents of `A`? Select all that apply.

```
Y y = new Y(); // in python: y = Y()
// an unknown sequence of operations
y.insert(1);
y.insert(2);
y.insert(3);
```

For technical reasons, `A` is given as a list below, using square brackets. Underscores `_` are considered empty in the data structure (i.e., they contain something that the data structure does not care about.)

- [x] a. `[ _, 1, 2, 3]` ✓
- [ ] b. `[ 2, 3, _, _, 1]`
- [x] c. `[ 2, 3, _, 1]` ✓
- [ ] d. `[ _, 1, 2, 3, _]`
- [ ] e. `[ 1, 2, _, 3]`

Question 8

Correct

Mark 1.00 out of 1.00

What are the values of `a` and `b` after executing the following piece of code?

Java:

```
Y y = newY();
Y z = newY();
Y w=z;
w.insert(3);
z.insert(1);
y.insert(2);
int a = z.remove();
int b = y.remove();
```

Python:

```
y = Y()
z = Y()
w = z
w.insert(3)
z.insert(1)
y.insert(2)
a = z.remove()
b = y.remove()
```

Write something like `a = 89 , b = 99`

Answer: a = 3 , b = 2 ✓

Question 9

Correct

Mark 1.00 out of 1.00

How many array accesses does a single call to `Y.remove` take in the worst case? ((To make this well-defined, we assume that the compiler performs no clever optimisations. That is, every array access we've written in the code will actually be performed.)

- a. 7
- b. 2 ✓
- c. ~4N
- d. ~2N

Question **10**

Correct

Mark 1.00 out of 1.00

How may array accesses does the most expensive public method (`insert` or `remove`) of `Y` take in the worst case. (Give the smallest correct answer.)

a. quadratic in N
b. linearithmic in N
c. linear in N ✓
d. constant

Question **11**

Correct

Mark 1.00 out of 1.00

What is the number of array accesses per operation in the following sequence of (2k) operations, starting from an empty data structure:

```
y.insert(1);
y.remove();
y.insert(2);
y.remove();
y.insert(3);
y.remove();
 . . .   // and so on
y.insert(k);
y.remove();
```

a. quadratic in k in the worst case
b. linear in k in the worst case and in the amortised sense
c. constant in the amortised sense, but linear in k in the worst case
d. constant in the worst case ✓

Question **12**

Complete

Marked out of 1.00

True or false: The data structure `Y` uses space linear in `N`. Explain you answer.

Be as formal and short as you can, but not shorter. If you use more than two paragraphs of text you’re on the wrong level of abstraction.

This is true as the array of A in class Y has a static length from the start, and its length is doubled when the number of elements in A is equal to the length of A.