【DS】Day4(2)

Tags	
Date	@May 25, 2022
Summary	Resizing Array Stack and Queue

【Week2】Stack

2.2 Resizing Arrays

If array is full, create a new array of twice the size, and copy items

```
public ArrayStack() {
  stack = new String[1];
}

public void push(String item) {
  if(N == stack.length)
    resize(2 * stack.length);
  stack[N++] = item;
}

private void resize(int capacity) {
  String[] copy = new String[capacity];
  for (int i = 0; i < N; ++i) {
    copy[i] = stack[i];
  }
  stack = copy;
}
```

To shrink the array:

Efficient solution:

- `pop():` Halve size of array stack[] when array is one-quarter full

```
public void pop() {
  String item = s[--N];
  s[N] = null;
  if(N > 0 && N == stack.length / 4)
    resize(stack.length / 2);
```

```
  return item;
}
```

2.3 Queues

APIs:

```
public class QueueOfStrings {
  QueueOfStrings() // Create an empty queue
  void enqueue(String item) // Insert a new string onto queue
  String dequeue() // Remove and return the string least recently added
  boolean isEmpty() // Is the queue empty?
  int size() // Number of strings on the queue
}
```

`dequeue()`:

```
public String dequeue() {
  String item = first.item;
  first = first.next;
  if(isEmpty())
    last = null;
  return item;
}
```

`enqueue()`:

```
public void enqueue(String item) {
  Node oldLast = last;
  last = new Node();
  last.item = item;
  last.next = null;
  if (isEmpty())
    first = last;
  else
    oldLast.next = last;
}
```