

**EXERCISE 2: Memory Analysis**

(a) Use tilde notation to describe how much memory an object of type `Grid<Item>` requires as a function of  $n$  right after the constructor finishes execution. Note that the grid is of size  $n \times n$ .

**Solution.**

Short answer:  $\sim 8n^2$  since there are  $n^2$  references to items in the grid.

Detailed answer:

- 16 Bytes (object overhead)
- 4 Bytes (int instance variable)
- 8 Bytes (reference to the array instance variable)
- 4 Bytes (padding)
- $8n + 24$  Bytes (array overhead +  $n$  references to arrays)
- $n \times (24 + 8n) = 24n + 8n^2$  Bytes ( $n$  arrays each containing  $n$  references to items)

$$\text{Total} = 16 + 4 + 8 + 4 + 8n + 24 + 24n + 8n^2 = 56 + 32n + 8n^2 = \sim 8n^2$$

(b) Use tilde notation to describe how much memory a `Grid<Integer>` object requires as a function of  $n$ , assuming that there are no *null* items in the grid. Note that every object of type `Integer` requires 24 Bytes.

**Solution.**

In addition to the  $n^2$  references to the items ( $\sim 8n^2$ ), we also have  $n^2$  items of type `Integer`. Therefore, the total memory required is:

$$8n^2 + 24n^2 = \sim 32n^2 \text{ Bytes.}$$

