**Circular Queue**

Previously, we have provided a straightforward but inefficient implementation of queue.

A more efficient way is to use a circular queue. Specifically, we may use a fixed-size array and two pointers to indicate the starting position and the ending position. And the goal is to reuse the wasted storage we mentioned previously.

Let's take a look at an example to see how a circular queue works. You should pay attention to the strategy we use to enqueue or dequeue an element.

A screenshot of a video

AI-generated content may be incorrect.

A screenshot of a video game

AI-generated content may be incorrect.

A screenshot of a video game

AI-generated content may be incorrect.

A screenshot of a video game

AI-generated content may be incorrect.

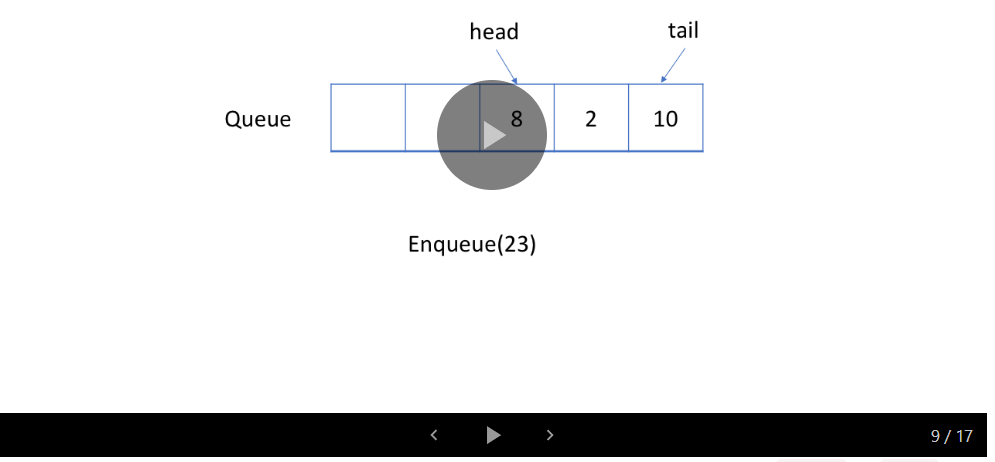
A screenshot of a video game

AI-generated content may be incorrect. A screenshot of a video game

AI-generated content may be incorrect. A screenshot of a video game

AI-generated content may be incorrect. A screenshot of a video game

AI-generated content may be incorrect.



A screenshot of a video game

AI-generated content may be incorrect.

A screenshot of a video game

AI-generated content may be incorrect.

A screenshot of a video game

AI-generated content may be incorrect.

A screenshot of a video game

AI-generated content may be incorrect.

A screenshot of a video game

AI-generated content may be incorrect.

A screenshot of a video game

AI-generated content may be incorrect.

A screenshot of a video game

AI-generated content may be incorrect.

A screenshot of a video game

AI-generated content may be incorrect.

1 / 17

Review the animation carefully to figure out the strategy we use to check if a queue is empty or full.

For the next exercise, we will let you try to implement the circular queue by yourself and provide a solution later.