Problem: Implement a priority queue backed by an array.

Constraints

- **Do we expect the methods to be insert, extract_min, and decrease_key?**
  - **Yes**
- **Can we assume there aren't any duplicate keys?**
  - **Yes**
- **Do we need to validate inputs?**
  - **No**
- **Can we assume this fits memory?**
  - **Yes**

Test Cases

insert

- **`insert` general case -> inserted node**

extract_min

- **`extract_min` from an empty list -> None**
- **`extract_min` general case -> min node**

decrease_key

- **`decrease_key` an invalid key -> None**
- **`decrease_key` general case -> updated node**

Algorithm

insert

- **Append to the internal array.**

Complexity:

- **Time: O(1)**
- **Space: O(1)**

extract_min

- **Loop through each item in the internal array**
    - **Update the min value as needed**
- **Remove the min element from the array and return it**

Complexity:

- **Time: O(n)**
- **Space: O(1)**

decrease_key

- **Loop through each item in the internal array to find the matching input**
    - **Update the matching element's key**

Complexity:

- **Time: O(n)**
- **Space: O(1)**