Mod 6 Lab - Modified Binary Search

First, some vocabulary:

- A list `L` is **monotonically decreasing** if $L[i] \leq L[i+1]$ for all `i`
- A list `L` is **monotonically increasing** if $L[i] \geq L[i+1]$ for all `i`

Note the equality operator in both definitions: a list where all items have the same value is both monotonically decreasing *and* monotonically increasing, because $L[i] = L[i+1]$ for all `i`.

We have seen that we can use a binary search to determine if a list contains an item in $O(\log \text{n})$ if that list is monotonically decreasing or monotonically increasing - this is what we typically mean by saying a list is "sorted."

Here, we will implement a binary search on lists that are "sorted" in a different way - our list with `n` items will be:

- monotonically decreasing, $L[: k+1]$
- monotonically increasing, $L[k :]$

where `k` is an integer and $0 < k < n$. Remember, python indices are "half open", so both bullet points above apply to the element at index k.

Additionally - there will be no repeats between the decreasing and increasing "halves" of the list. A noteable exception is the minimum value which is by definition included in both halves.

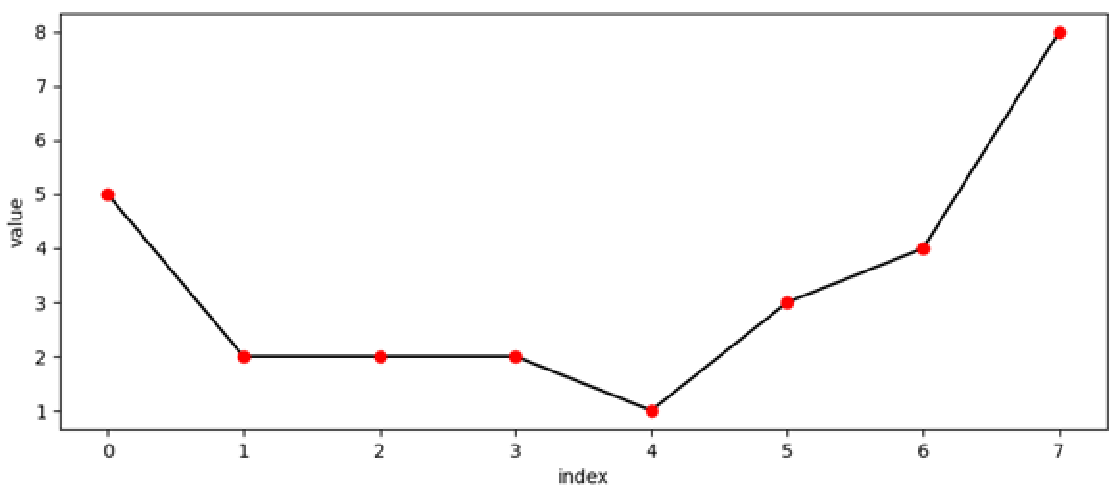

Figure 1: Plot of value vs index for sample list `L = [5, 2, 2, 2, 1, 3, 4, 9]`

Deliverable - `find_min`

Write a function `find_min` that finds the minimum item in a list sorted as described above.

- `find_min(L, m)`
  - $O(\text{logn})$
  - Returns the smallest item in a list `L` sorted as described above. Return the value, not the index.
  - `m` is the maximum number of times an item in the list repeats. Remember, there are no duplicate values on the increasing and decreasing halves (except for the minimum value).

Submission

At a minimum, submit the following files:

- `lab6.py`

Students must submit to Mimir **individually** by the due date (typically, Sunday at 11:59 pm EST) to receive credit.

Grading

You will be graded based on functionality and running time.

- 25 - `find_min` with `m == 1`
- 25 - `find_min` with `m == 1`, $O(\log n)$
- 25 - `find_min` with `m > 1`
- 25 - `find_min` with `m > 1`, $O(\log n)$

Feedback

If you have any feedback on this assignment, please leave it here.

We check this feedback regularly, and it has resulted in many improvements.