Explanation

that implementation could be sent in memory, since the priority queue already works in memory and does not need a cache. The way to get the time could be through BigO notation

The consultation cost is O (1)

**Max-Heapify** (*A*, *i*):

*left* ← 2×*i*     *// ← means "assignment"*

*right* ← 2×*i* + 1

*largest* ← *i*

**if** *left* ≤ *heap\_length*[*A*] **and** *A*[*left*] > A[*largest*] **then**:

*largest* ← *left*

**if** *right* ≤ *heap\_length*[*A*] **and** *A*[*right*] > *A*[*largest*] **then**:

*largest* ← *right*

**if** *largest* ≠ *i* **then**:

**swap** *A*[*i*] and *A*[*largest*]

**Max-Heapify**(*A*, *largest*)

<?php

  echo "Example Initial for queue!!";

  $pq = new SplPriorityQueue();

  // The insert method inserts an element in the queue by shifting it up

  $pq->insert('A', 3);

  $pq->insert('B', 6);

  $pq->insert('C', 1);

  $pq->insert('D', 2);

  // Count the elements

  echo "count ->" . $pq->count() . PHP\_EOL;

  // Sets the mode of extraction (EXTR\_DATA, EXTR\_PRIORITY, EXTR\_BOTH)

  $pq->setExtractFlags(SplPriorityQueue::EXTR\_BOTH);

  // Go at the node from the top of the queue

  $pq->top();

  // Iterate the queue (by priority) and display each element

  while ($pq->valid()) {

      print\_r($pq->current());

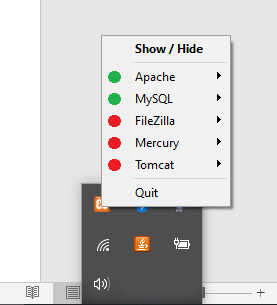
      echo PHP\_EOL;

      $pq->next();

  }

?>

Running this local server for example I used the xampp:



Open the new url in your browser for example:

<http://localhost/test_giant_monkey_robot/>

