CSCI 136
Data Structures & Advanced Programming

Heapsort: an *in-place* `O(nlogn)` sort

Video Outline

- Heapsort
  - Description
  - Comparison to Quicksort
  - When to use heapsort?

HeapSort

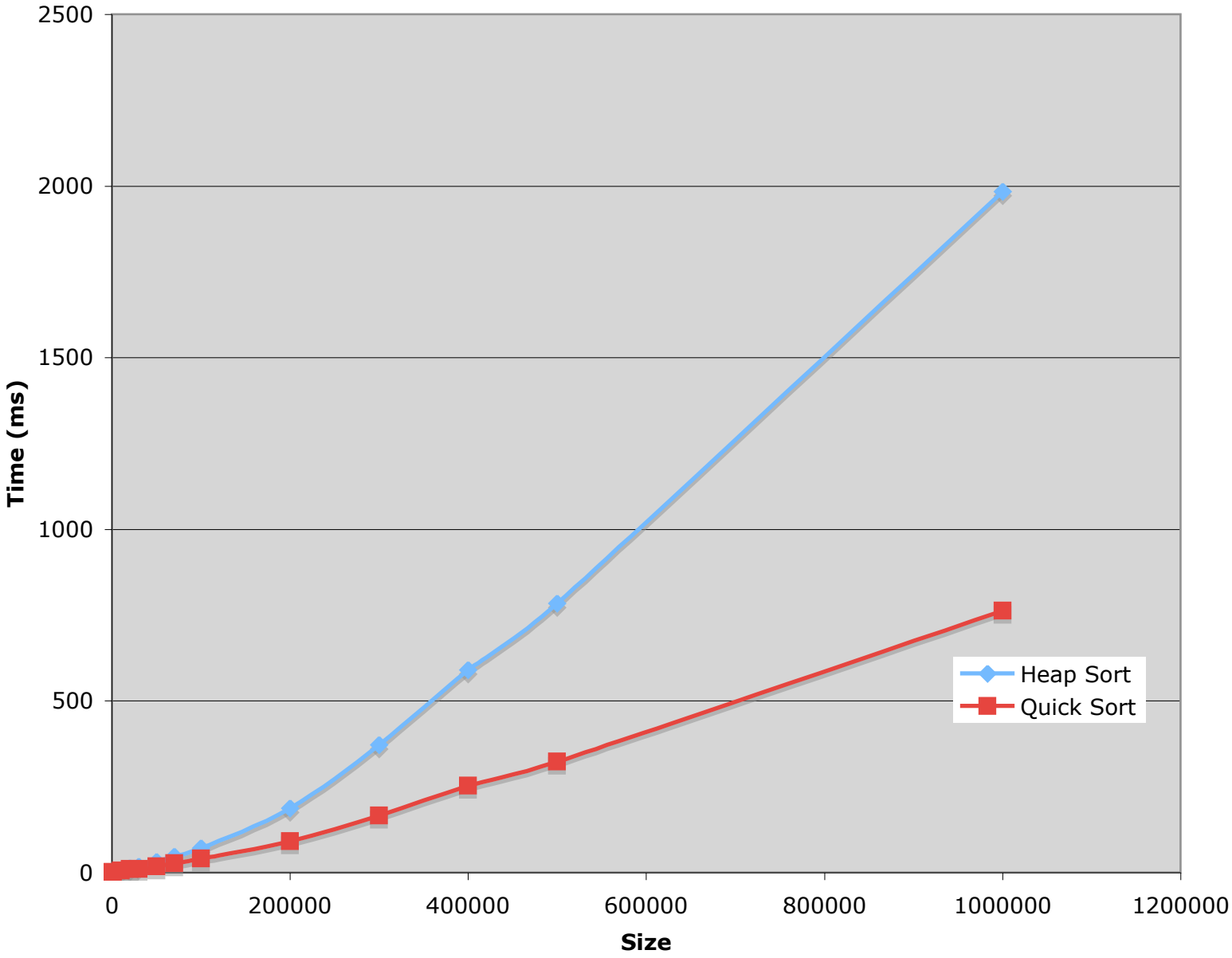
- Kind of an "Advanced" version of Selection Sort
- Strategy:
  1. Make a *max-heap*: array[0…n]
     - array[0] is largest value
     - array[n] is rightmost leaf
  2. Take the largest value (array[0]) and swap it with the rightmost leaf (array[n])
  3. Call pushDownRoot on array[0…n-1]
     - Now our "heap" is one element smaller, and the largest element is at end of array.

  Repeat until heap is empty and array is sorted

HeapSort

- Another O(n log n) sort method
- Heapsort is not *stable*
  - The relative ordering of elements is not preserved in the final sort
    - Why not?
      - There are multiple valid heaps given the same data
- Heapsort can be done *in-place*
  - No extra memory required!!!
  - Great for resource-constrained environments

Heap Sort vs QuickSort



Why Heapsort?

- Heapsort is slower than Quicksort in general
- Any benefits to heapsort?
  - *Guaranteed* O(n log n) runtime
- Works well on mostly sorted data, unlike quicksort
- Good for incremental sorting