Taylor Earl

10/22/14

Cs1030

* 1. Arrays
     1. Student Grades
     2. Phone Numbers
  2. Stacks
     1. Storing a value in memory
     2. Performing a mathematical calculation
  3. Queue
     1. Print queues
     2. Download queues
  4. Timsort
     1. Pros
        1. Designed to work well on many kinds of real world data
        2. Takes advantage of partial orderings that already exist to make it more efficient
     2. Cons
        1. Only stable if presented in a strictly descending order
  5. Comb Sort
     1. Pros
        1. Based off of the slower bubble sort, but quicker
        2. The gap that it uses to sort with can be greater than 1
     2. Cons
        1. Its performance isn't as good as originally expected
        2. Slower than its brother, the shell sort
  6. Radix sort
     1. Pros
        1. Measured in number comparisons, not actual time complexity
     2. Cons
        1. Hard to determine the actual efficiency

1. <https://www.cs.usfca.edu/~galles/visualization/ComparisonSort.html>
   1. The insertion sort basically takes two items and compares them and moves it down until its the bigger item, then it moves on and does all of the items.