CS2302 - Data Structures
Summer 2018
Lab 1
Due Wednesday, June 20, 2018

The median of a list or array $A$ is the element $a$ of $A$ such that half of the elements in $A$ are smaller than $a$ and half of them are larger than $a$. For example, the median of array $A = \{20, 10, 45, 1, 12\}$ is 12, since 12 is greater than 1 and 10 and smaller that 20 and 45.

An easy way to find the median is to sort the array or list and return the element in the middle:

```java
class MedianFinder {
    public int median(int[] A) {
        sort(A);
        return A[A.length / 2];
    }
}
```

Your task for this lab is to implement several algorithms for finding the median of a list of integers, using elements of type iNode as described in class, and compare their running times for various list lengths. To generate data to test your methods, write a method that receives an integer $n$ and builds and returns a list of random integers of length $n$.

The algorithms to compare are the following:

1. Sort list using bubble sort, then return the element in the middle.
2. Sort list using merge sort, then return the element in the middle.
3. Sort list using quicksort, then return the element in the middle.
4. Implement a modified version of quicksort that makes a single recursive call instead of the two made by normal quicksort, processing only the sublist where the median is known to reside.

Write a report describing your work. For every method, determine the big-O running time with respect to $n$. Run experiments with various values of $n$ to determine if their analytical running times agree with what you see in practice (notice that unless $n$ is large, running times will be too small to draw any meaningful comparisons). Illustrate your experimental results using tables and/or plots. Do not include in your time computations the time required to generate the list or to output the results. Also, make sure that for a given array all 4 algorithms return the same value.