







Homework 2

This problem will help you remember the idea of implementing Comparable in java (read SW 1.4)

Problem 1 [20 points]. (*Die.java*) Implement a data type Die in Die.java that represents a six-sided die and supports the following API:

method	description
<code>Die()</code>	constructs a die
<code>void roll()</code>	rolls the die
<code>int value()</code>	returns the face value of the die
<code>boolean equals(Die that)</code>	returns true if this and that die have the same face value, and false otherwise
<code>String toString()</code>	returns a string representation of the current face value of the die, ie,  ,  ,  ,  ,  , or 

```
$ java Die 3 3 5
* *
*
* *
false
true
```

These problems will help you understand the Analysis of Algorithms (read CLRS)

Problem 1. [5 points] (True or False): No comparison-based sorting algorithm can do better than $\Omega(n \log n)$ in the worst-case

Problem 2 [10 points]. Prove $o(g(n)) \cap \omega(g(n))$ is the empty set. [note: $o \rightarrow$ little-oh and $\omega \rightarrow$ little omega)

Problem 3.[25 points] Consider functions $f(n)$ and $g(n)$ as given below. Use the most precise asymptotic notation to show how function f is related to function g in each case (i.e. , $f \in ?(g)$). For example, if you were given the pair of functions $f(n) = n$ and $g(n) = n^2$ then the correct answer would be: $f \in o(g)$. To avoid any ambiguity between $O(g)$ and $o(g)$ notations due to writing, use *Big- $O(g)$* instead of $O(g)$.

Homework 2

$f(n)$	$g(n)$	Relation
$n^3 + 2n + 1$	$\frac{1}{100}n^3 + n \log n$	
2^n	n^{1000}	
$\log_2 n$	$\log_3 n$	
2^n	3^n	
0.5^n	1	

Problem 4.[10 points] In each of the pairs of the functions below, **verify** using the definition of “big Oh”, whether $f(n) = O(g(n))$, $g(n) = O(f(n))$ or both.

(a) $f(n) = 20n^2 + 14n + 7$ $g(n) = 50n$

(b) $f(n) = n^2$ $g(n) = 2^n$

Submitting Information:

- Use the code I provided for each problem. DON'T DELETE ANY FUNCTION
- You should have each problem in a separate .java files (ex: Die.java, hw2.doc).
- Submit your work on Canvas in one Zip file.
- The deadline is Wednesday, Feb 26th at 11.55PM