Homework #1

*Please present a **printed** report with all your answers, explanations, and sample plots. Submit also a soft copy of the source code and binaries used to generate these results. Please note that copying of any results or source code will result in ZERO credit for the whole homework.*

Problem 1

Is $2^{n+1}=O(2^n)$? Is $2^{2n}=O(2^n)$? Justify your answers.

Problem 2

For each of the following relationships, find two non-negative functions $f(n)$ and $g(n)$ that satisfy it. Justify your answers.

1. $f(n)=O(g(n))$ and $g(n)=O(f(n))$
2. $f(n)=O(g(n))$ and $g(n)\neq O(f(n))$
3. $f(n)=\Theta(g(n))$ and $g(n)=\Omega(f(n))$

Problem 3

Give an asymptotic upper and lower bounds for *T(n)* for each of the following recurrences. Justify your answers. Assume that *T(n)* is constant for $n < 10$.

1. $T(n)=3T(n/3)+n$
2. $T(n)=3T(n/3)+n^2$
3. $T(n)=11T(n/7)+n^3$
4. $T(n)=16T(n/2)+\binom{n}{3}\log^4 n$
5. $T(n)=T(n-10)+n$
6. $T(n)=T(n/3)+T(n/4)+n$

Problem 4

Describe a $\Theta(n\log n)$ -time algorithm that, given a set *S* of *n* integers and another integer *x*, determines whether or not there exists two elements in *S* whose elements sum to *x*. Analyze its running time to verify it takes $\Theta(n\log n)$.

For example, a C++ function should look like this:

```
bool check_S(const vector<int>& S, int x) {
```

```
    . . .
}
```

Problem 5

Write two C++ functions that sort an input array of integers (in place) using Heapsort and Quicksort (described in the textbook).

1. Run your algorithm on a random array of sizes 5, 10, 25, 50, 100, 500, 1000, and 10000. Measure the CPU time for each such run for each algorithm, and plot the results. To get more accurate timings, you should take the average of 25 runs for each input array size (with a random array each time).
2. Repeat part (1) above but on sorted arrays i.e. generate a random array of integers, and sort it first (both in increasing and decreasing order) and plot the running time of both Heapsort and Quicksort. Which one is better?
3. Modify your Quicksort to choose a *random* pivot instead of the last element of the array, and
4. Repeat parts (1) and (2). Does it improve?

Instructions

Please submit a soft copy of the solutions together with source code and binaries in one zip file. The file should be named as **CMP448.HW##.BN##.First.Last.zip** where HW## is the homework number e.g. HW01, BN## is your bench number, First and Last are your first and last name. So, if your bench number is 26, your name is Mohamed Aly, the file should be named **CMP448.HW01.BN26.Mohamed.Aly.zip.** Failing to follow these instructions will cost you points.

Acknowledgment: Some problems are adapted from Erik Demaine.