CS 223
Laboratory Assignment #1 (8 marks)
Due: at the end of lab in Week 4 (Feb 6, 09)

Design efficient algorithms with few nested loops

In this first laboratory exercise, you will design, implement, and analyze different algorithms for the following problems.

Problems

1. Suppose we are given 2 arrays of real numbers of length n=10000 each, A[9999], B[9999], how can we find $a \in A, b \in B$ such that $a = b$ (and announce the negative result if no such a, b exist)? You can easily design an $O(n^2)$ time algorithm for this problem. But can you do better? Compare the actual running time of these two programs. If the difference is not obvious, increase n.

2. The following problem is called 3SUM and is very famous in algorithm design. Given 3 arrays of reals of length n=5000 each, A[4999], B[4999], C[4999], how can we find $a \in A, b \in B, c \in C$ such that $a + b = c$?

It is not difficult to write a program to solve this problem in $O(n^3)$ time. But can you obtain an $O(n^2)$ time solution? You might want to initiate A, B, C with random numbers, say, with the following loop

```
for (i = 0; i< n-1; i++) {
     A[i] = rand()%n;
     B[i] = rand()%n;
     C[i] = rand()%(33*n);
     }
```

Again, compare the actual running time of these two programs. If the difference is not obvious, increase n until the difference becomes noticeable.

Solve recurrence relations

3. Solve the following recurrence relations and prove your claim by induction. In call cases $T(1) = 1$.

3.1) $T(n) = 2T(n/2) + \log n$.

3.2) $T(n) = 8T(n/2) + n^3$.

3.3) $T(n) = T(n/4) + 2n^2$.

3.4) $T(n) = T(n-1) + 3n - 1$.