Introduction To Algorithms
CS430

Spring 2013
HomeWork 1
Due 28th January

1. **Exercise 1.2-3 on page 14 (CLRS 3rd Edition).**

   What is the smallest value of n such that an algorithm whose running time is $100n^2$ runs faster than an algorithm whose running time is 2^n on the same machine.

2. **Problem 2** Consider a ternary search algorithm similar to binary search algorithm, except that instead of a comparing with a single array element in each iteration we now have 2 elements at postions $\frac{1}{3}$ and $\frac{2}{3}$ distance from the begining of the list. The pseudo code is given below. Draw out the decision tree for a list of 10 elements and analyze the worst case time of the algorithm as a function n, the number of elements in the list.

Algorithm 1 $Ternary - search(A[], key, min, max)$

```
while $max \geq min + 2$ do
  $mid1 = min + \lfloor \frac{max-min+1}{3} \rfloor$
  $mid2 = min + \lfloor \frac{2(max-min+1)}{3} \rfloor$
  if $A[mid1] > key$ then
    $max = mid1 - 1$, i.e. work on $A[min] \ldots A[mid1 - 1]$
  else if $A[mid1] == key$ then
    return mid1
  else
    if $A[mid1] < key < A[mid2]$ then
      $min = mid1 + 1, max = mid2 - 1$, i.e. work on $A[mid1] \ldots A[mid2 - 1]$
    else if $A[mid2] < key$ then
      $min = mid2 + 1$, i.e. work on $A[mid2 + 1] \ldots A[max]$
    else
      return $mid2$
    end if
  end if
end while
Compare $A[min] \ldots A[max]$ with $key$ and return result.
```