# Solutions Part 4
# Discrete Mathematics

Hans Georg Schaathun

10th August 2015

## 1 Thursday 2 October

**Exercise 1.1** *Consider an array $A$ of objects, where each object $o$ has a key $k(o)$. Consider an algorithm which takes a search key $k_0$ as input and outputs an element $o \in A$ so that $k_0 = k(o)$.*

1. *Write pseudo-code for a search algorithm. How many objects must be considered before the right element is found? Give answers for the worst case and the average case.*
2. *Suppose the array $A$ is sorted with keys $k(o)$ in increasing order. How does that affect searching? Write pseudo-code for a faster algorithm, taking advantage of the search order. How many objects must now be considered in the worst case before the right element is found?*

**Exercise 1.2** *Draw a recursion tree for*

$$T(n) = 3(T(n/3)) + n, \quad \text{when } n \geq 2, \tag{1}$$

$$T(1) = 1. \tag{2}$$

*Assuming that $n$ is a power of three, use the recursion tree to find an exact solution for $T(n)$.*

**Exercise 1.3** *Draw a recursion tree for*

$$T(n) = 4(T(n/4)) + 3n, \quad \text{when } n \geq 1, \tag{3}$$

$$T(1) = 1. \tag{4}$$

*Assuming that $n$ is a power of four, use the recursion tree to find an exact solution for $T(n)$.*

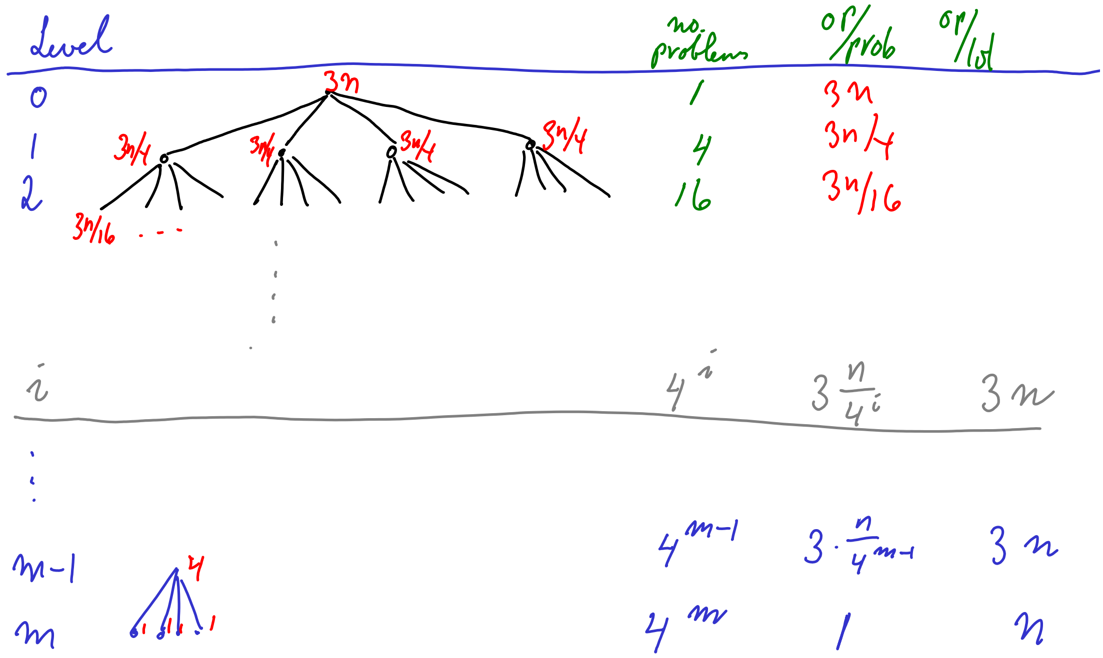


Figure 1: Recurrence tree for Problem 1.3.

**Solution:** The recurrence tree is shown Figure 1, where $n = 4^m$. We see $m$ levels of $3n$ operations each, plus one level of $n$ operations. Hence $T(n) = 3n \log n + n$, assuming that $n$ is a power of 4.

**Exercise 1.4** *Revisit the recursion trees from Exercises 1.2 and 1.3. Use the diagrams to find Big-O bounds on the two recurrence equations.*

# 2 Friday 3 October