Practice 2 – Solving recurrences

Exercise 1

Solve the following recurrences to get the best asymptotic bounds you can on T(n) in each case using O notation:

a. T(n) = $T\left(\frac{n}{4log_2n}\right) + 2n$ for n > 1 and T(1) = 1. You can assume that all numbers are rounded down to the nearest integer.
b. BFPRT algorithm for median finding: $T(n) \leq cn + T\left(\frac{n}{5}\right) + T(\frac{3n}{4})$ and T(1) = 1, where c ≥ 1. You can assume that everything is rounded down to the nearest integer
c. Polynomial Multiplication: $T(n) = 3T\left(\left\lceil\frac{n}{2}\right\rceil\right) + cn$ and T(1) = 4 .
d. Matrix Multiplication: $T(n) = 7T\left(\left\lceil\frac{n}{2}\right\rceil\right) + cn^2$ and T(1) = 1.
e. $T(n) \leq T\left(\left\lceil\frac{n}{2}\right\rceil\right) + T\left(\left\lfloor\frac{n}{2}\right\rfloor\right) + cn\log n$, for n ≥ 4 and T(2) = 2.

Exercise 2

Consider the following algorithm:

Input: A number x and integer n ≥ 0
Output: The value x^n

```
POWER(x, n)
if n = 0 then
      return 1
if n is odd then
      y ← POWER(x, (n – 1)/2)
      return x · y · y
else
      y ← POWER(x, n/2)
      return y · y
```

a. Write down the recurrence equation which describes the running time of POWER(x, n) as a function of n.
b. Assuming that n is a power of 2, solve this recurrence equation using either the substitution method or the recursion tree technique. Give the asymptotic (Big-Oh) complexity of POWER(x, n).

Exercise 3

Solve the following recurrences using the Master theorem when applicable:

a. $T(n) = 2T(n/2) + n^3$
b. T(n) = T(9n/10) + n
c. $T(n) = 16T(n/4) + n^2$

d. $T(n) = 7T(n/3) + n^2$
e. $T(n) = 7T(n/2) + n^2$
f. $T(n) = 2T(n/4) + \sqrt{n}$
g. $T(n) = T(n - 1) + n$
h. $T(n) = T(\sqrt{n}) + 1$