Homework#2 Fundamentals

Textbook:

1.17. Find two monotonically increasing functions $f(n)$ and $g(n)$ such that $f(n) \neq O(g(n))$ and $g(n) \neq O(f(n))$.

1.34. Write an algorithm to find the maximum and minimum of a sequence of n integers stored in array $A[1..n]$ such that its time complexity is

(a) $O(n)$.
(b) $\Omega(n \log n)$.

1.35. Let $A[1..n]$ be an array of integers, where $n > 2$. Give an $O(1)$ time algorithm to find an element in A that is neither the maximum nor the minimum.

1.37. Give an algorithm that evaluates an input polynomial

$$a_n x^n + a_{n-1} x^{n-1} + \ldots + a_1 x + a_0$$

for a given value of x in time

(a) $\Omega(n^2)$.
(b) $O(n)$.

Egg drop. Suppose that you have an *N*-story building and plenty of eggs. An egg breaks if it is dropped from floor *T* or higher and does not break otherwise. Your goal is to devise a strategy to determine the value of *T* given the following limitations on the number of eggs and tosses:

- Version 0: 1 egg, $\leq T$ tosses.
- Version 1: log*N* eggs and log*N* tosses.
- Version 2: log*T* eggs and 2log*T* tosses.
- (Advanced)Version 3: 2 eggs and 2$\sqrt{N}$ tosses.
- (Advanced)Version 4: 2 eggs and $\leq c\sqrt{T}$ tosses for some fixed constant *c*.