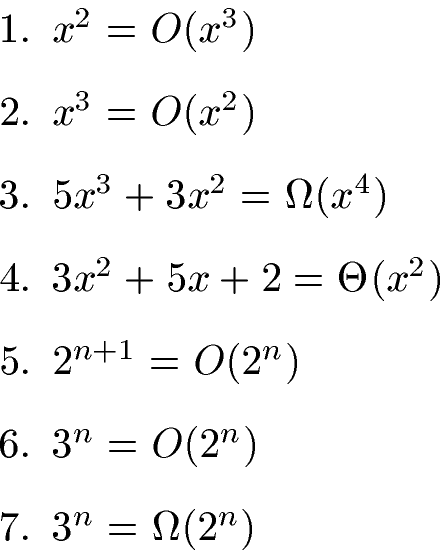
Algorithm Analysis

Handout 2

Deadline October 5

**Exercise 2.1**

For each of the following equations show whether it is true or false:

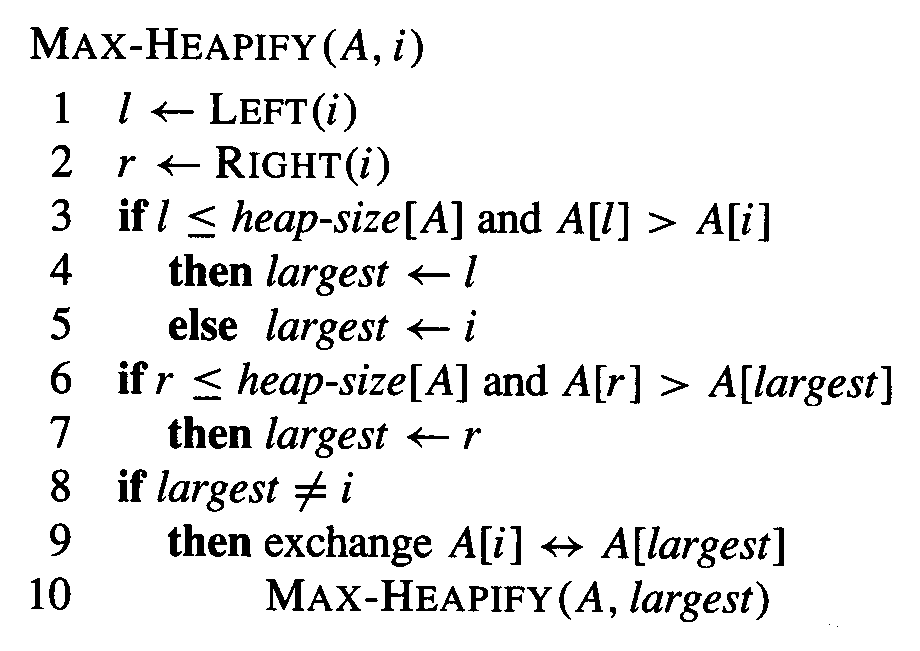


**Solutions:**

1. : We use and and get: For all :
2. : We consider that implies . Such cannot exist because will always become larger than any constant. Therefore, the **statement is false**.
3. : We consider that implies . Such cannot exist because the term will always become smaller than any constant for increasing . Therefore, the **statement is false**.
4. : We require two constants (for upper bound) and (for lower bound) as well as the value . First, we rewrite to . For , the term evaluates to and for increasing it becomes smaller than 10 but never falls below 3. Therefore, we can choose the two asymptotic constants as follows: and . For we use 1 and get:  
    and for all .
5. : We use and and get: for all .
6. : We consider that implies . Such cannot exist because the term becomes larger than any constant for increasing . Therefore, the **statement is false**.
7. : We use and and get: for all (our .

# Exercise 2.2

The Max-Heapify algorithm in your textbook has a recursive definition:



Give an iterative (loop – based) definition of the Max-Heapify algorithm.

**Solution:**

Pseudocode of iterative Max-Heapify:

MAX-HEAPIFY(A, i)

**while** (true)

l = LEFT(i)

r = RIGHT(i)

**if** l<=heap-size[A] and A[l]>A[i]

then largest = l

else largest = i

**if** r<=heap-size[A] and A[r]>A[largest]

then largest = r

**if** largest!=i

then exchange (A[i], A[largest])

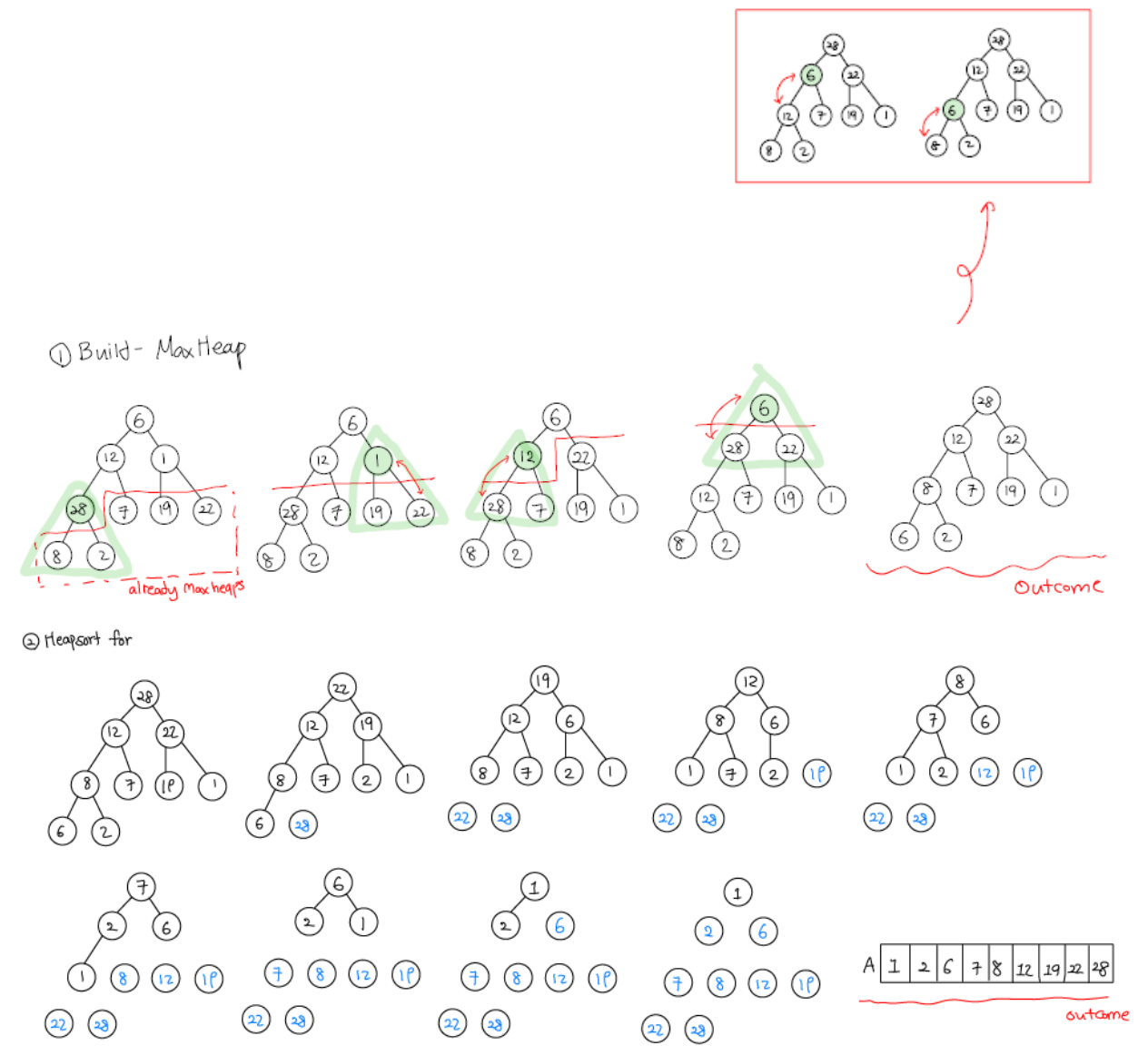
i = largest

else break // while loop

# Exercise 2.3

Using Figure 6.4 on page 161 in the textbook as a model, illustrate the operation of Heap sort on the array **A = [6, 12, 1, 28, 7, 19, 22, 8, 2]**

**Solution:**

(The below figure is copied from the homework report of a participating student of the fall semester course 2020)