Exercise 2. Answer Sheet

Student's Name: Haruki Terushima Student's ID: s1290037

Problem 1. *(10 points)* Consider a priority queue S implemented as a heap. Write a pseudo-code for the **Maximum(S)** operation on this priority queue.

```
def Maximum(S)
    if S[1] > S[2]:
        maximum = S[1]
    else:
        maximum=-∞
        for i =A.length/2 to A. length
            if maximum<S[i]:
                maximum= S[i]
            end
        end
    end
    return maximum
end
```

Problem 2. *(20 points)* Consider top-down heap construction approach.

a). Write a pseudo-code for a **HeapTopDown(A)** algorithm using **Max-Heap-Insert (A, key)** operation

```
def HeapBottomUp (A)
    //Input: An array A[1..n] of orderable items
    //Output: A heap A[1..n]

    A. heapsize = A. length
    for i =1 to A. length
        Max-Heap-Insert (A, A[i])
    end
end
```

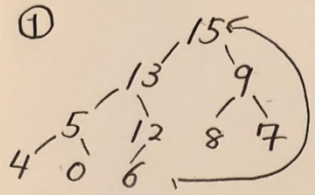
b) What is the time complexity of **HeapTopDown(A)** algorithm? Why?

Log(n*logn)

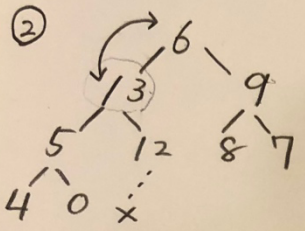
Problem 3. *(20 points)* Illustrate the operation **Heap-Extract-Max** on a heap A=[15,13,9,5,12,8,7,4,0,6]

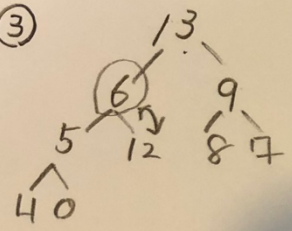
Problem 3

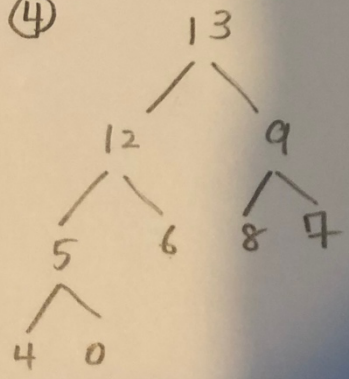
Heap - Extract - Max [15 13 9 5 12 8 7 4 0 6]

① 

max = 15

② 

③ 

④ 

⑤ return max = 15

Problem 4. *(50 points)* Write a program implementing **HeapBottomUp (A)** algorithm. Upload your source code. Show your input array and the output heap in the space below.

python3 HeapBottomUp.py

```
input
10
4
1
3
2
16
9
10
14
8
7

output
[16, 14, 10, 9, 8, 7, 4, 3, 2, 1]
```