

Exercise 2. Answer Sheet

Student's Name: Tsuyoshi Kumamoto

Student's ID:s1250050

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.

Put your answer here.

```
def Maximum(S)
    return S[1]
```

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

Put your answer here.

```
Def Max-Heap-Insert(A,k)
    A.heap_size = A.heap_size + 1
    A[A.heap_size] = MAX_NEGATIVE
    Heap-Increase-Key(A, A.heap_size, k)
```

```
Def Heapsort
    HeapTopDown(A)
    Max-Heap-Insert(A,k)
    For I = A.length to 2
        Swap(A[1],A[i])
        A.heap_size = A.heap_size -1
        Max Heapify(A,1)
```

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

Put your answer here.

Time complexity: **$O(\log n)$**

For loop (repeated **$n - 1$** times) which has time complexity **$O(\log n)$** .

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

Put your answer here.

15

13,12

9,8,7,6

5,4,0

```
def Extract-Max (A)
    // Input: heap A[1..n]
    // Removes and returns the root element
    max = A[1]
    A[1] = A[A.heap_size]
    A.heap_size = A.heap_size - 1
    MaxHeapify (A, 1)
    return max
```

import A[1](15) to max = A[1]
Next, write A[1] from A[A.heap_size] . A[1] = 0
Finally, reconfigure the heap

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.A[]

Put your answer here.

```
#include <stdio.h>
#include <stdlib.h>

void swap(int *x, int *y){
    int t;

    t = *x;
    *x = *y;
    *y = t;
}

void bottomUp(int A[], int first, int last){
    int parent = first;
    int child = parent*2;

    while(child <= last){
        if((child < last)&&(A[child] < A[child+1])){
            child++;
        }
        if(A[child] <= A[parent]){
            break;
        }
    }
}
```

```

        swap(&A[child], &A[parent]);

        parent = child;
        child = parent*2;
    }
}

void heapSort(int A[], int size){
    int i;

    for(i=size/2; i>=1; i--){
        buttomUp(A, i, size);
    }

    for(i=size; i>=1; i--){
        swap(&A[i], &A[1]);
        buttomUp(A,1,i-1);
    }
}

int main(){
    int A[7];
    int i;

    printf("Please input 7th Numbers\n");
    for(i=0; i<7; i++){
        scanf("%d",&A[i]);
    }

    heapSort(A,7);

    for(i=0; i<7; i++){
        printf("%d ",A[i]);
    }
    printf("\n");

    return 0;
}

```

```

1
4
5
3
7
8
9
1 3 4 5 7 7 8

```