Data Structures

Homework #3

Fallen leaves

Heap, Heap’, Heap’’

□ **Objective**: write a C++ program for constructing a heap (max heap) and printing it in three different forms.

□ **Input**: heap operations of size *n* from the file

□ **Output**: the final shape of the constructed heap in given three forms

□ Note that *n* ranges from 1 to 150

Input Domain & Operations

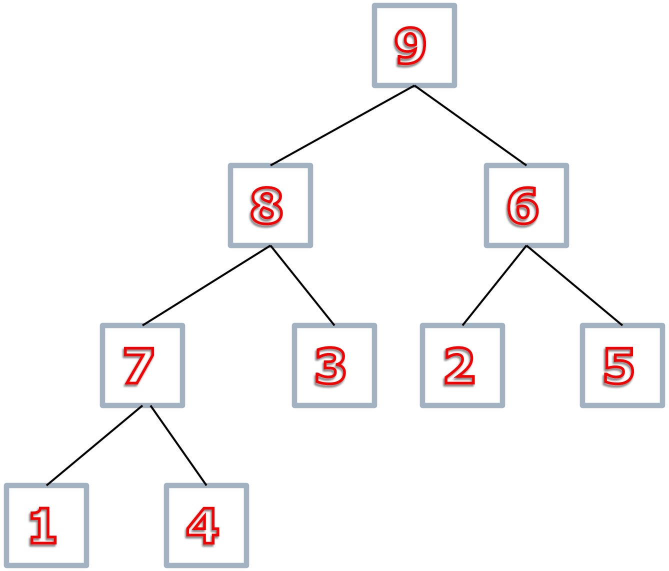
- **Only the following 62 characters are inserted:**
  - 0 < 1 < 2 < 3 < 4 < 5 < 6 < 7 < 8 < 9 < a < b < c < d < e < f < g < h < i < j < k < l < m < n < o < p < q < r < s < t < u < v < w < x < y < z < A < B < C < D < E < F < G < H < I < J < K < L < M < N < O < P < Q < R < S < T < U < V < W < X < Y < Z
- **Input operations**
  - "I #": insert a character '#' into the heap
  - "D": delete the root node from the heap

Input Format (File Input)

- **Read from a text file**
  - Input file name: **input.txt**
    - The first line: a positive integer *n*
    - The second line: *n* heap operations
    - Example

```
9
I 1 I 2 I 3 I 4 I 5 I 6 I 7 I 8 I 9
```

Constructed Heap for Example Input

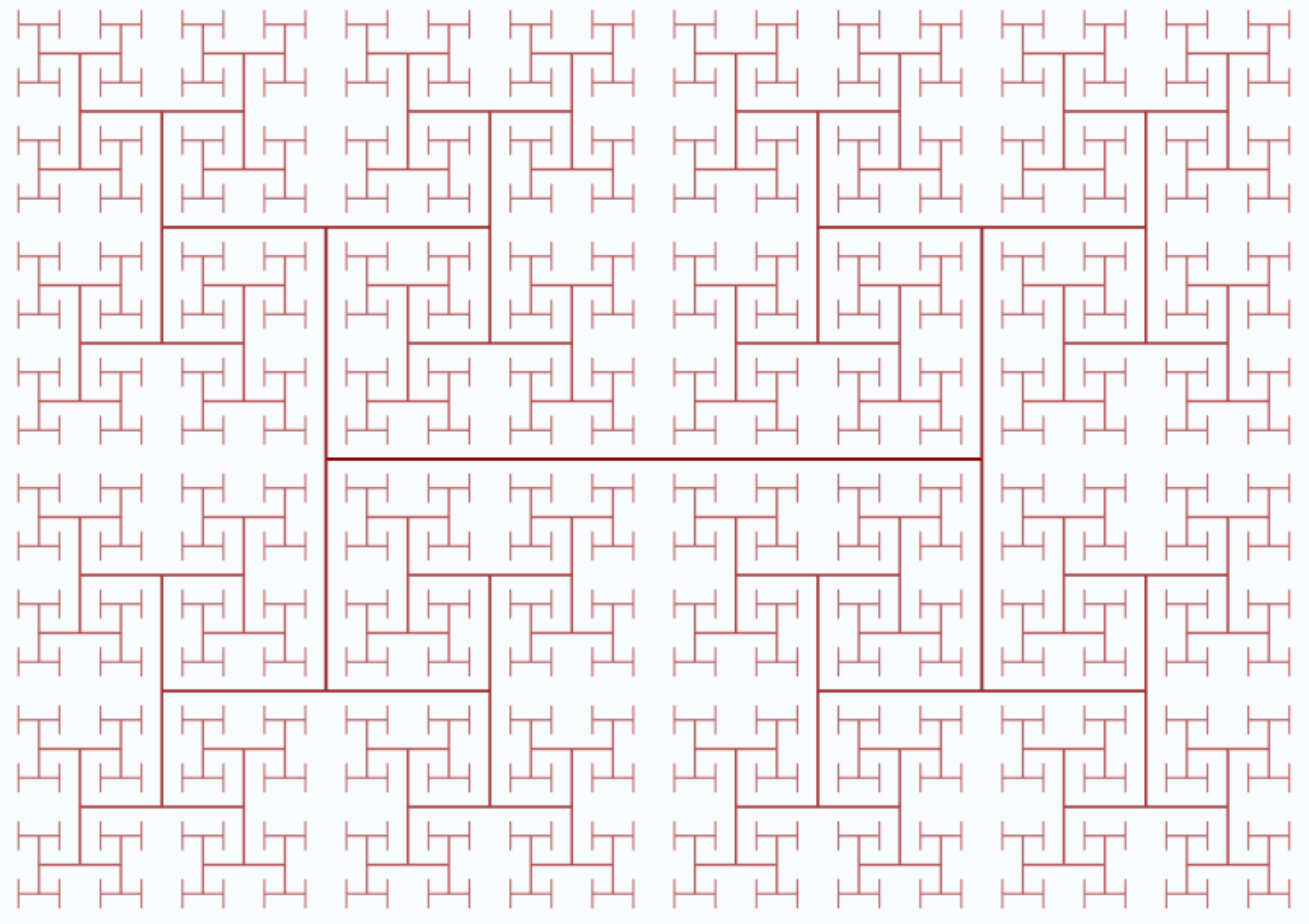


Output Format (Console Output)

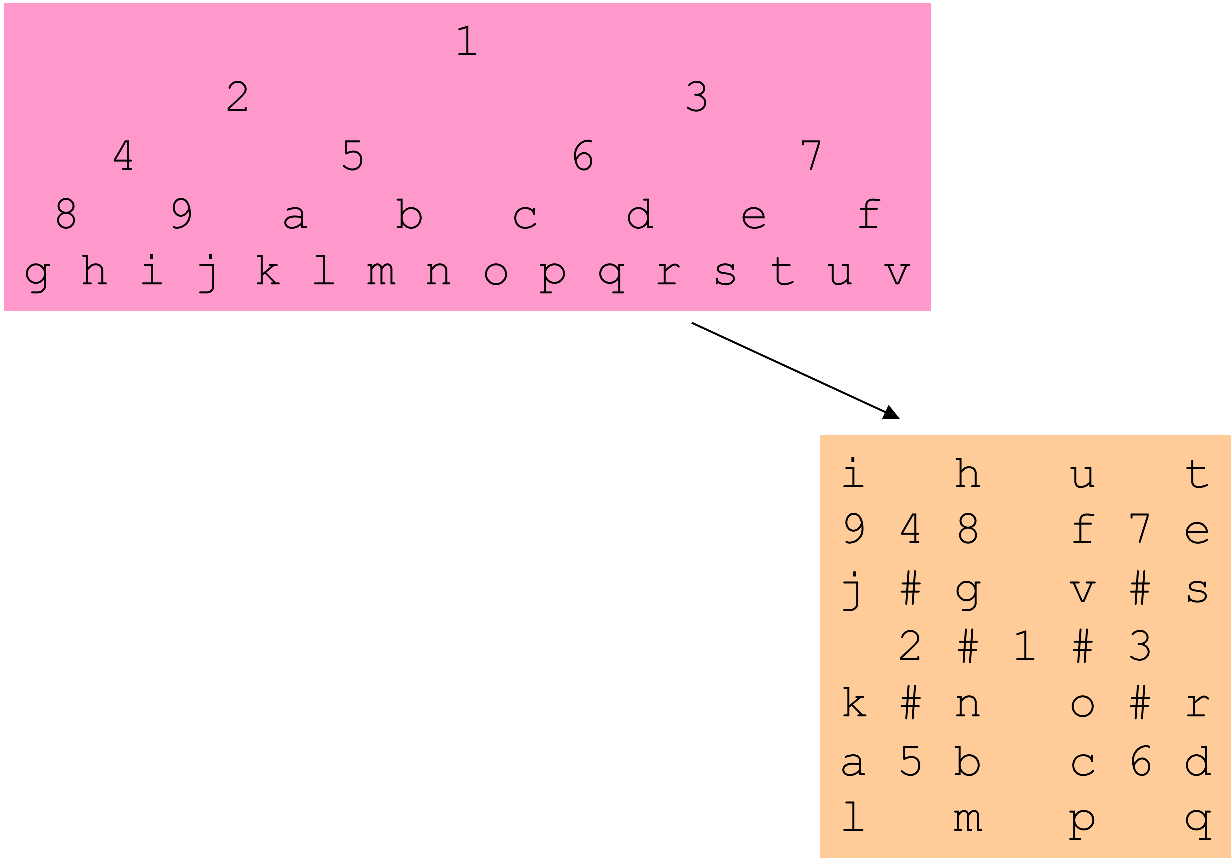
- Print “`1. rotated form\n`” and then print the constructed heap after rotating the heap by counterclockwise 90 degree.
  - The width of the value of each node should be 2*depth.

- Print “`2. not-rotated form\n`” and then print the constructed heap as it is.
  - The width of the value of each leaf node should be 1.

- Print “`3. H-tree form\n`” and then print the constructed heap as a H-tree form.
  - See next three slides.



H-tree Form



```
               1
       2               3
   4       5       6       7
 8   9   a   b   c   d   e   f
g h i j k l m n o p q r s t u v
```

```
i   h   u   t
9 4 8   f 7 e
j # g   v # s
  2 # 1 # 3
k # n   o # r
a 5 b   c 6 d
l   m   p   q
```

Hint: Pseudo-code for Printing H-tree

```
str[] = "123456789ABCDEFGHIJKLMNOPQRSTUV";
V[4][2] = {{-1, 0}, {1, 0}, {0, 1}, {0, -1}};
H(node, i, j, d, U, D, R, L)
{
    if(node > n) return;
    H_tree[i][j] = str[node-1];
    if(2*node <= n) {
        H_tree[i + d*V[L][0]][j + d*V[L][1]] = str[2*node-1];
        H(4*node, i + d*(V[L][0]+V[U][0]),
            j + d*(V[L][1]+V[U][1]), d/2, D, U, L, R);
        H(4*node+1, i + d*(V[L][0]+V[D][0]),
            j + d*(V[L][1]+V[D][1]), d/2, U, D, R, L);
    }
    if(2*node+1 <= n) {
        H_tree[i + d*V[R][0]][j + d*V[R][1]] = str[2*node];
        H(4*node+2, i + d*(V[R][0]+V[D][0]),
            j + d*(V[R][1]+V[D][1]), d/2, U, D, R, L);
        H(4*node+3, i + d*(V[R][0]+V[U][0]),
            j + d*(V[R][1]+V[U][1]), d/2, D, U, L, R);
    }
}
center(n) { return n <= 1 ? 0 :2 * center(n/4) + 1; }
depth(n) { return n <= 7 ? 1 : 2 * depth(n/4); }
CALL  H(1, center(n), center(n), depth(n), N, S, E, W);
```

Output for Example Input

```
1. rotated form
    5
  6
    2
9
    3
  8
      4
    7
      1
```

```
2. not-rotated form
       9
   8          6
 7   3   2   5
1 4
```

```
3. H-tree form
4 7 1     5
  8 # 9 # 6
  3       2
```

Due Date

- **Soft deadline: November 30, 2017**
- **Hard deadline: December 3, 2017**
  - But, will deduct 20% per one day from your original score.

Submission Date	Deduction Rate
December 1	20%
December 2	40%
December 3	60%
December 4	100%

Notice (cont'd)

- **You should observe the format of input & output exactly.**

- **You should submit a compressed file (HW3_your-ID.zip) containing the following three files to the u-campus web-site (http://info.kw.ac.kr).**
  - **HW3_your-ID.hwp/doc** // report document
  - **HW3_your-ID.cpp** // source code
  - **HW3_your-ID.exe** // executable file

Notice (cont'd)

- **Source code**
  - It should be compiled in **Visual Studio 2010 or higher, or g++**
    - **You should note your environment in your report.**
  - Your name and student ID should be noted at the top of your source files in the form of comment.
- **Report**
  - Free format
  - But, it must include several examples for testing your program and your own discussion.
  - It will be an important factor for getting a good score.