Data Structures 2022

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.

- **Console input**: heap operations of size n
- **Console output**: the final shape of the constructed heap in given three forms

- Note that n ranges from 1 to 200

Input Domain & Operations

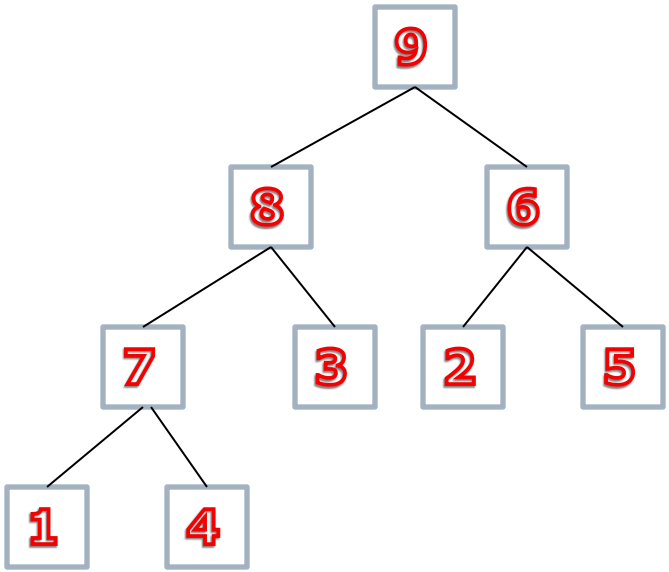
- **Only the following 63 characters are inserted:**
  - `0 < 1 < 2 < 3 < 4 < 5 < 6 < 7 < 8 < 9 < ? < A < B < C < D < E < F < G < H < I < J < K < L < M < N < O < P < Q < R < S < T < U < V < W < X < Y < Z < a < b < c < d < e < f < g < h < i < j < k < l < m < n < o < p < q < r < s < t < u < v < w < x < y < z`
  - Input operations
    - "`INS #`": insert a character '#' into the heap
    - "`DEL`": delete the root node from the heap
    - "`EOI`": end of input

Input Format (Console Input)

- **Read from keyboard**
  - Example

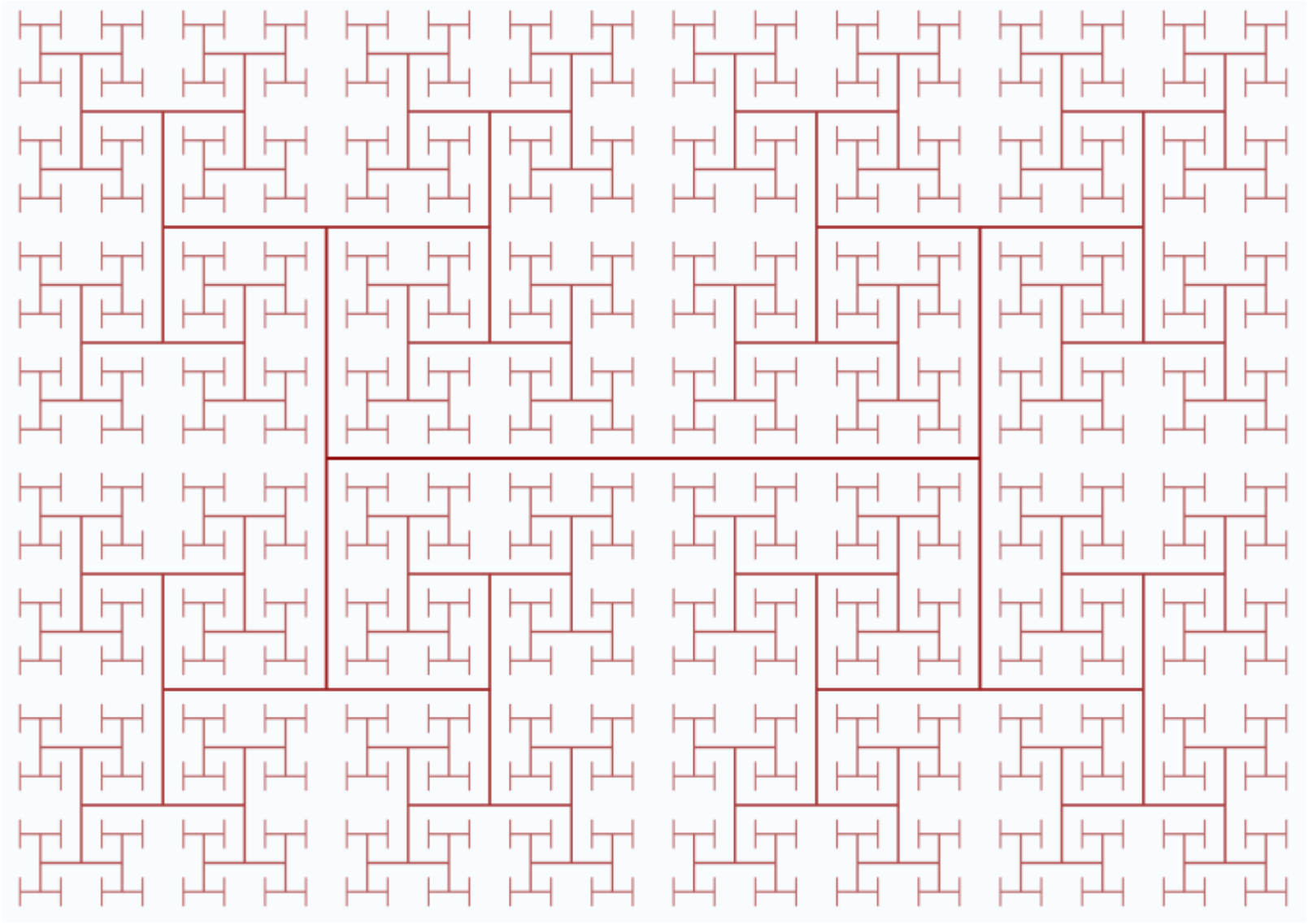
```
INS 1 INS 2 INS 3 INS 4 INS 5
INS 6 INS 7 INS 8 INS 9 EOI
```

Constructed Heap for Example Input

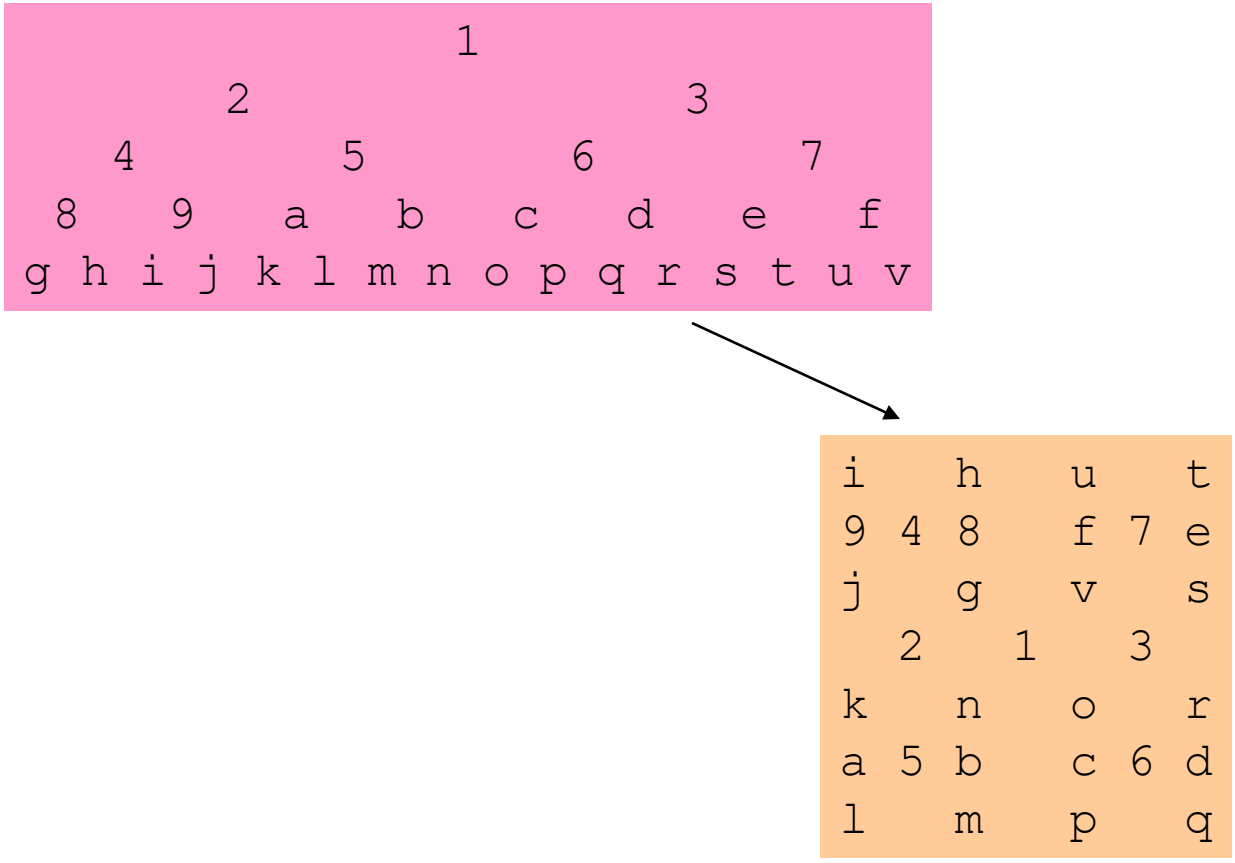


Output Format (Console Output)

- 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: December 6, 2022**
- Hard deadline: December 8, 2022
  - But, will deduct 20% per one day from your original score.

Submission Date	Deduction Rate
December 7	20%
December 8	40%
December 9	100%

Notice (cont'd)

- You must observe the format of input & output exactly.
- For heap operations, you must use "algorithm" in STL.
  - E.g., push_heap(…) and pop_heap(…)

- You have to submit a compressed file (**HW3_your-ID.zip**) containing the following two files to the web-site (https://klas.kw.ac.kr).
  - **HW3_your-ID.hwp**, **-.doc**, or **-.pdf** // report document
  - **HW3_your-ID.cpp** or **-.cc** // source code

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.
  - Mention your compiler.