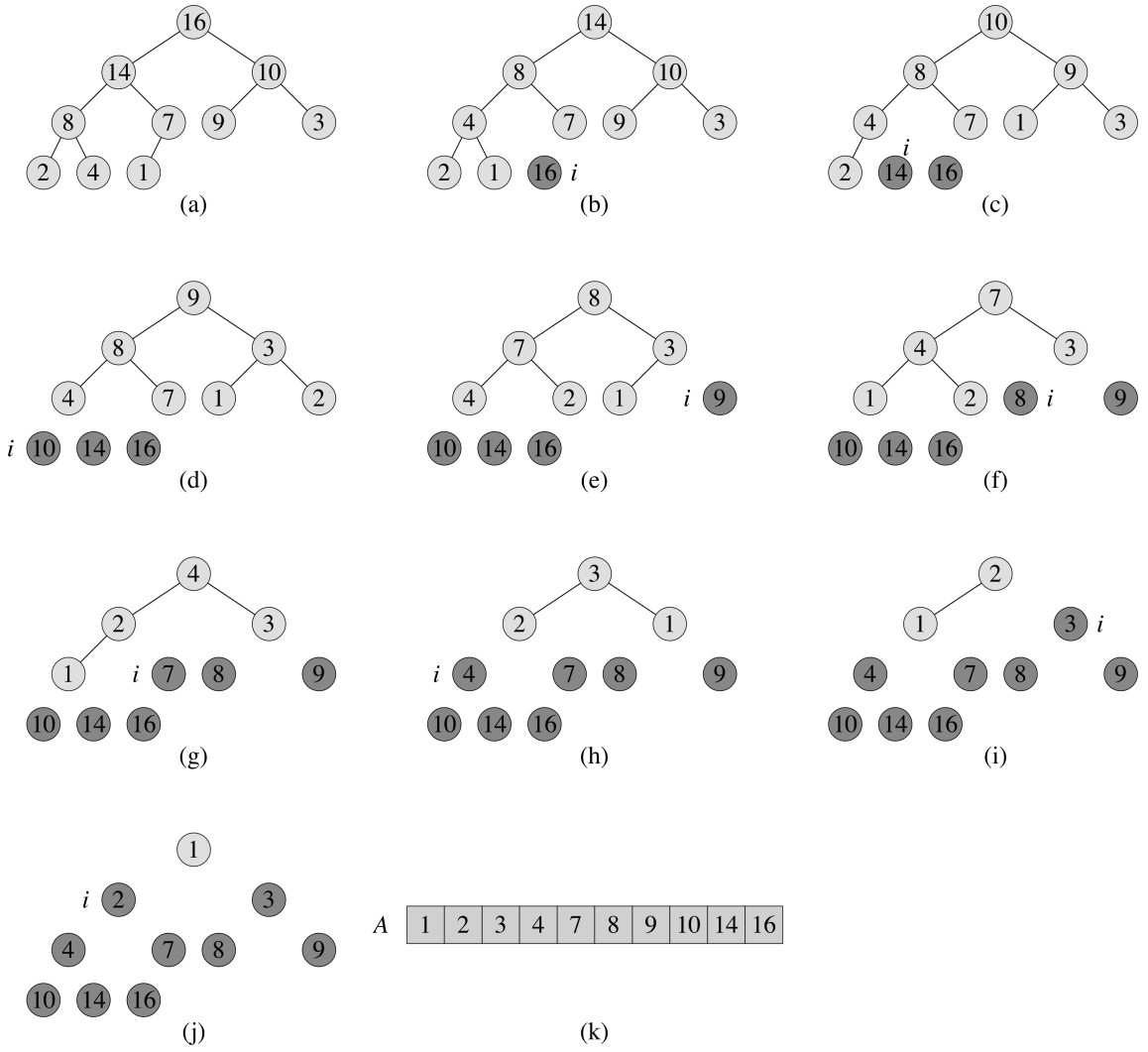
Heapsort (Max-heap implementation)

Summary: In this lab you will implement the heapsort algorithm. The pseudo code is provided below. Your program must use an array to store the sequence of data to be sorted. The array must begin at index 1. Your method names must match the pseudo code. Call them `Heapsort`, `BuildMaxHeap`, and `MaxHeapify`. The return type for a particular method and the number and the type of the arguments for that method are an engineering choice made by you based on your individual Java implementation. A graphical example of the running of the algorithm is provided. Your entire implementation must be in a single class file named `heapsort.java`.

HEAPSORT(A, n)
  BUILD-MAX-HEAP(A, n)
  for $i = n$ **downto** 2
    exchange $A[1]$ with $A[i]$
    MAX-HEAPIFY$(A, 1, i - 1)$



NOTE: Refer to the CLRS course textbook pp. 159-161 for further explanation of heapsort.

File I/O
Your program will be tested on two files. The name of these two files will be `input1.txt` and `input2.txt`. The format will be the same. The number of elements in the input sequence is on the first line followed by the input sequence on the second line. The first file's contents are below.

```
input1.txt:
10
16  14  10  8  7  9  3  2  4  1
```

For testing, your program must write the contents of the heap instances to the console in the following format.

```
input1.txt:
16  14  10  8  7  9  3  2  4  1
14  8  10  4  7  9  3  2  1  16
10  8  9  4  7  1  3  2  14  16
9  8  3  4  7  1  2  10  14  16
8  7  3  4  2  1  9  10  14  16
7  4  3  1  2  8  9  10  14  16
4  2  3  1  7  8  9  10  14  16
3  2  1  4  7  8  9  10  14  16
2  1  3  4  7  8  9  10  14  16
1  2  3  4  7  8  9  10  14  16
```

To demo for your TA, invoke your program with the following command:
`%java heapsort input1.txt`

Demo: Demo your working code for your TA.
Submission: Submit your `heapsort.java` file via turn-in link on Piazza. If you do not finish and demo in your lab session, create and submit a tar archive named `lab5.tgz` that also includes a `README` text file that clearly states what functionality your program successfully implements and what is missing or not working properly. Do not forget to put a class header on each your submission's files.

Note: Your submission must be either `heapsort.java` or `lab5.tgz`. See above. No other files will be accepted.

Rubric:
20 pts Attendance (On-time, attend entire lab)
80 pts Perfect functionality
-15 pts Minor errors/bugs in code such as partially incomplete methods and/or incorrect output.
-15 pts Major missing functionality or significant errors/bugs