CS21003 ALGORITHMS-1
(Tutorial 7: Heaps, Linear-time sorting)
Date: Oct 31 2020

1 Ugly Numbers

Ugly numbers are numbers whose only prime factors are 2, 3 or 5. The sequence $1, 2, 3, 4, 5, 6, 8, 9, 10, 12, 15, \ldots$ shows the first 11 ugly numbers. By convention, 1 is included. Given a number n, the task is to find n^{th} ugly number. Find an efficient algorithm to do so. What would be the complexity?

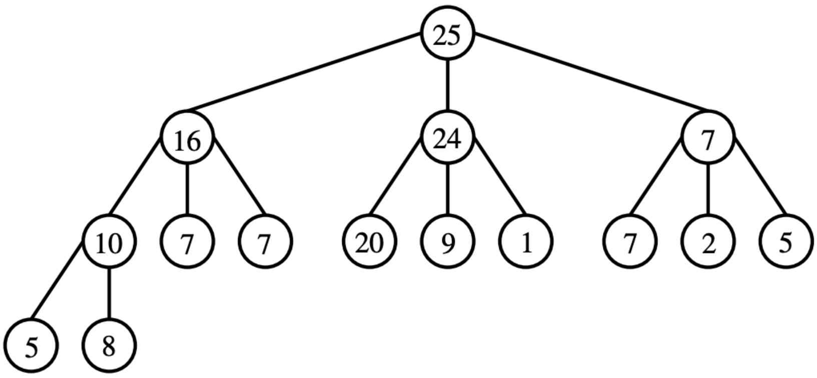
2 Median for a stream of data

You are given that integers are read from a data stream. Give an algorithm to find the median of elements read so far in an efficient way.
For example, let us consider the stream $5, 15, 1, 3, \ldots$

- After reading 1st element of stream: $5 \rightarrow$ median $= 5$
- After reading 2nd element of stream: $5, 15 \rightarrow$ median $= 10$
- After reading 3rd element of stream: $5, 15, 1 \rightarrow$ median $= 5$
- After reading 4th element of stream: $5, 15, 1, 3 \rightarrow$ median $= 4$, so on...

3 Ternary Heaps

Heaps introduced in the class are called binary heaps. *Ternary heaps* are realized by complete ternary trees where each node can have three children: left, mid, and right. All levels except perhaps the last must be full. Leaves appear only in the last two levels. Also, the leaves at the last level must be to the left of the empty positions. Finally, the key stored at any node must not be smaller than the keys stored in its children. The figure below illustrates a ternary heap and its contiguous representation.



25	16	24	7	10	7	7	20	9	1	7	2	5	5	8

Propose an algorithm to convert an array H of integers of size n to a ternary heap in the contiguous representation. More precisely, write the *heapify* and *makeheap* functions for a ternary heap.

4 Linear-time Sorting

- Extend the counting sort algorithm so that it works for both positive and negative numbers
- You are given an array of n dates in the $dd - mm - yyyy$ format. Propose a linear-time algorithm to sort the array in the usual increasing order (chronological order).
- Suppose that n points are chosen uniformly inside a circle of radius r (that is, the probability of choosing a point in any region R of area a inside the circle is $a/(\pi r^2)$). Give an algorithm that sorts the n given points with respect to their distances from the center of the circle in expected linear time.

5 Priority Change

Let us add the facility to a priority queue that the priority of an item may change after insertion. Provide an algorithm *changePriority* that, given the index of an element in the supporting array and a new priority value, assigns the new priority value to the element, and reorganizes the array so that heap ordering is restored. Your algorithm should run in $O(logn)$ time for a heap of n elements .

6 Maximum Sum Combination

Given two equal sized arrays (A, B) and N (size of both arrays), a sum combination is made by adding one element from array A and another element of array B. Display the maximum K valid sum combinations from all the possible sum combinations. What is the most efficient algorithm you can think of?