> *I thought the following four [rules] would be enough, provided that I made a firm and constant resolution not to fail even once in the observance of them. The first was never to accept anything as true if I had not evident knowledge of its being so.... The second, to divide each problem I examined into as many parts as was feasible, and as was requisite for its better solution. The third, to direct my thoughts in an orderly way... establishing an order in thought even when the objects had no natural priority one to another. And the last, to make throughout such complete enumerations and such general surveys that I might be sure of leaving nothing out.*
>
> — René Descartes, *Discours de la Méthode* (1637)

4 Randomized Binary Search Trees (February 1)

4.1 Treaps

In this lecture, we will consider binary trees where every internal node has both a *search key* and a *priority*. In our examples, we will use letters for the search keys and numbers for the priorities. A *treap* is a binary tree where the inorder sequence of search keys is sorted and each node's priority is smaller than the priorities of its children.[1] In other words, a treap is simultaneously a binary search tree for the search keys and a (min-)heap for the priorities.