COMP3711: Design and Analysis of Algorithms

Tutorial 6

HKUST

Question 1

Let X be a random variable that is equal to the number of heads in two flips of a fair coin. What is $E[X^2]$? What is $E^2[X]$?

Solution 1

X can be $0, 1, 2$ and thus X^2 can be $0, 1, 4$.

$$
\begin{aligned}
E[X^2] &= 0 \cdot Pr(X^2 = 0) + 1 \cdot Pr(X^2 = 1) + 4 \cdot Pr(X^2 = 4) \\
&= 1/2 + 1 \\
&= 3/2
\end{aligned}
$$

$$
\begin{aligned}
E[X] &= 0 \cdot Pr(X = 0) + 1 \cdot Pr(X = 1) + 2 \cdot Pr(X = 2) \\
&= 1/2 + 1/2 \\
&= 1
\end{aligned}
$$

So, $E^2[X] = 1^2 = 1$

Question 2

Explain why the worst-case running time for bucket sort is $\Theta(n^2)$. What simple change to the algorithm preserves its linear average-case running time and makes its worst-case running time $O(n \log n)$?

Solution 2

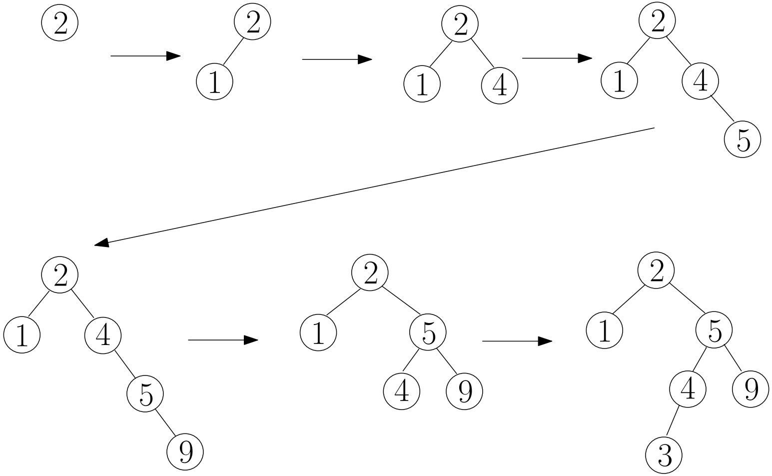
The worst-case running time for bucket sort is $\Theta(n^2)$ because this is possible that there are cn elements fall in the same bucket for some constant $0 < c < 1$, and thus the running time to sort the elements in that bucket by insertion sort is $\Theta(n^2)$.

Simply use any sorting algorithm with worst-case running time $O(n \log n)$ (e.g. merge sort) to replace insertion sort. This can preserve the linear average-case running time and makes the worst-case running time $O(n \log n)$.

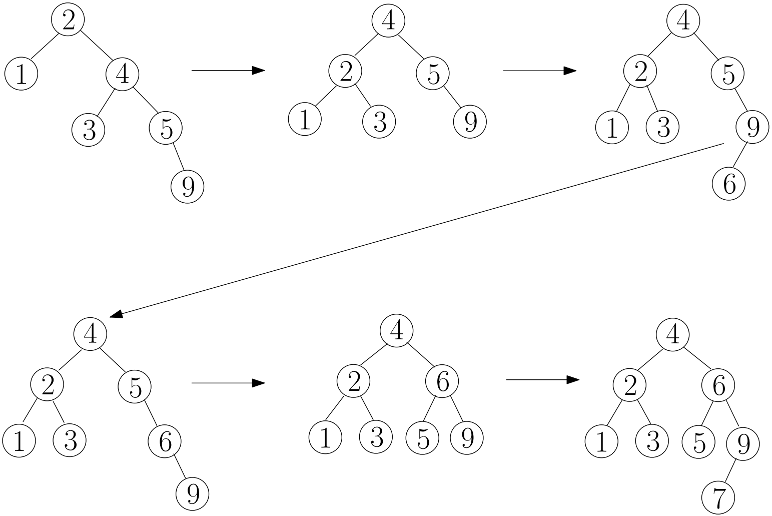
Question 3

Show the steps of inserting 2,1,4,5,9,3,6,7 into an initially empty AVL tree.

Solution 3



Solution 3



Question 4

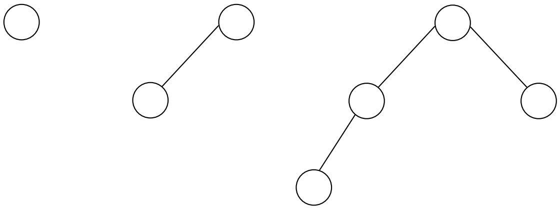
The AVL tree maintains its $O(\log n)$ height by balancing the heights of every two siblings. It is also possible to do so by balancing the weights. More precisely, the *weight* of a node u, denoted as $w(u)$, is the number of nodes in the subtree below u (including u). The weight of an empty tree is 0. We use u_L and u_R to denote the weight of u's left and right child, respectively. A node u is said to be *weight-balanced* if

$$\frac{1}{2} \leq \frac{w(u_L) + 1}{w(u_R) + 1} \leq 2.$$

A binary tree is weight-balanced if all of its nodes are weight-balanced.

Question 4

(a) The following figures show the smallest weight-balanced trees of height $0, 1$, and 2, respectively. Please draw the smallest weight-balanced trees of height 3 and 4.

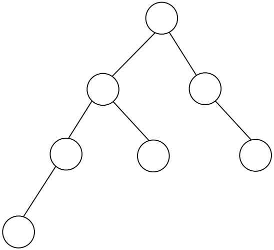


$n_0 = 1$ $n_1 = 2$ $n_2 = 4$

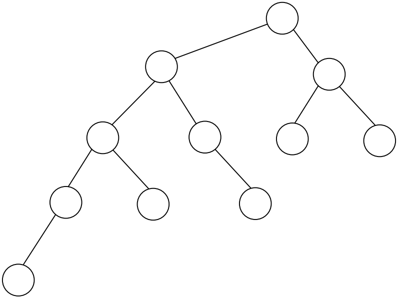
(b) Show that the height of a weight-balanced binary tree with n nodes is $O(\log n)$.

Solution 4

(a) The trees of height 3 and 4:



$n_3 = 7$



$n_4 = 11$

Solution 2

(b) Let n_h be the smallest size of a weight-balanced tree of height h. We have
$n_h \geq n_{h-1} + \frac{n_{h-1}+1}{2} - 1 + 1 = \frac{3}{2}n_{h-1} + \frac{1}{2} > \frac{3}{2}n_{h-1}$. Solving the recurrence yields $n_h > (\frac{3}{2})^h$, so $h = O(\log n)$.