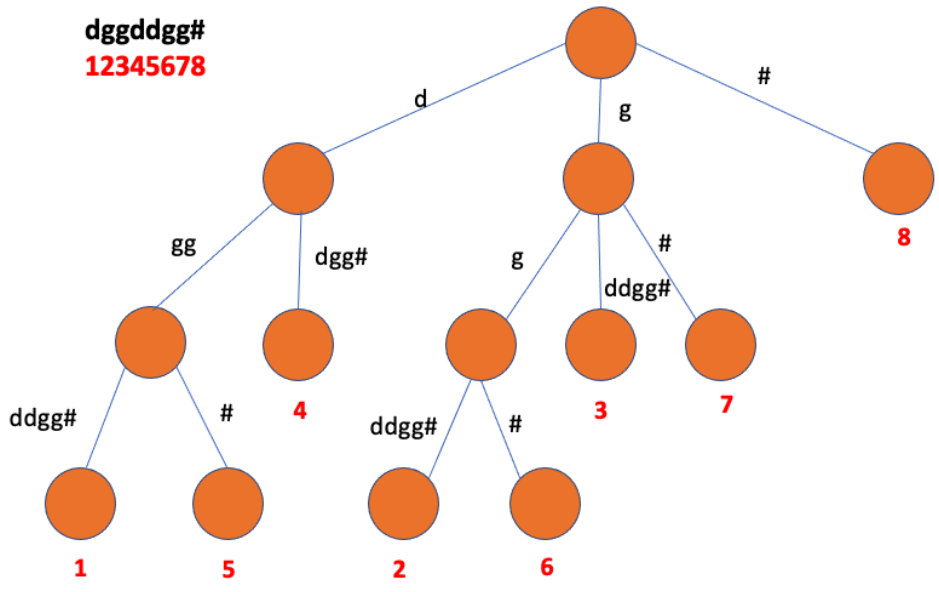
COP 5536 Fall 2020 Assignment 2

Liweichang | UFID: 67379542

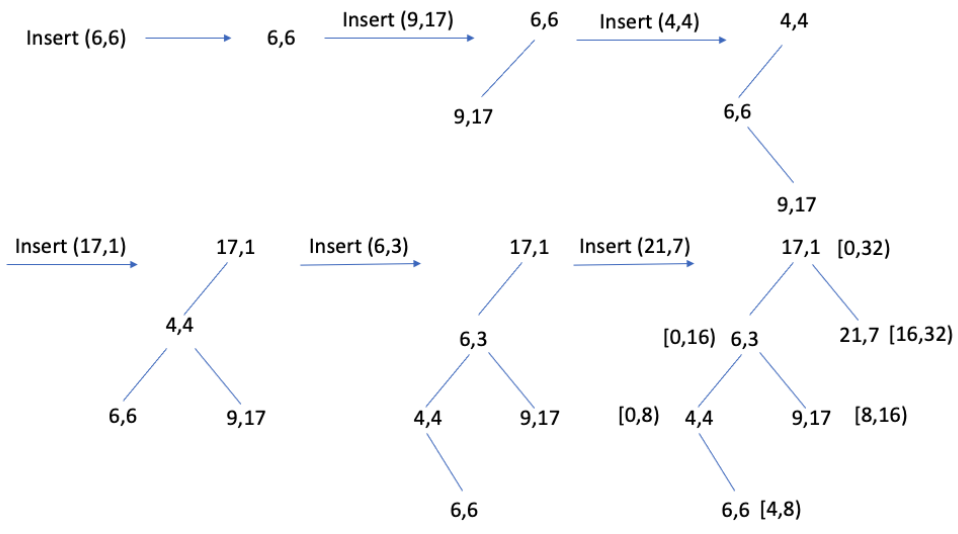
Q1: (a)



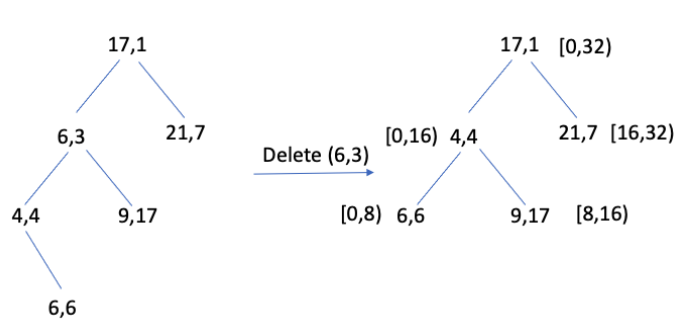
(b) : For two strings **S; T**,

1. create a generalized string **U = S$T#**.
2. construct suffix tree for the generalized string **U** in $O(|S| + |T|)$. Each leaf of the tree represents either a suffix from one of the two strings or a suffix that occurs in both the strings
3. Mark each internal node with a **s(t)** if there is a leaf in the subtree of v representing a suffix from **S (T)**. The path-label of any internal node marked both **s** and **t** is a substring common to both S1 and S2, and the longest such sting is the longest common substring.
4. So we find the node with the greatest string-depth that is marked both **s** and **t**. The node marking and calculations of string-depth can be done by standard linear-time tree traversal methods

Q2: (a)



(b)

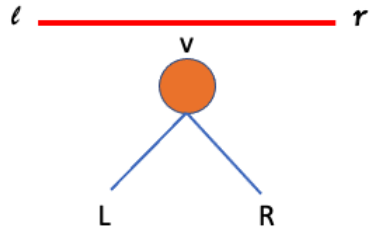


Q3: (a) n = 100000 is large, $h = (\ln 2)m/u \sim 0.693 * (5000/1000) = 3.47$, so h = 3 or 4.

(b) h = 3 → $p(u) = (1 - 1/n)^u * \left[1 - (1 - 1/m)^{uh}\right]^h = (1 - 1/100000)^{1000} * (1 - (1 - 1/5000)^{3000})^3 \sim 0.09095$

h = 4 → $p(u) = (1 - 1/n)^u * \left[1 - (1 - 1/m)^{uh}\right]^h = (1 - 1/100000)^{1000} * (1 - (1 - 1/5000)^{4000})^4 \sim 0.09106$

Q4:



1. Query interval is $[l, r]$;
2. if v.key ∈ $[l, r]$, all intervals in v overlap and then search L; R for additional overlapping intervals.

3. if v.key < l, intervals in v with r_i >= l overlap and then search R for additional overlapping intervals.
4. if v.key > r, intervals in v with l_i <= r overlap and then search L for additional overlapping intervals.