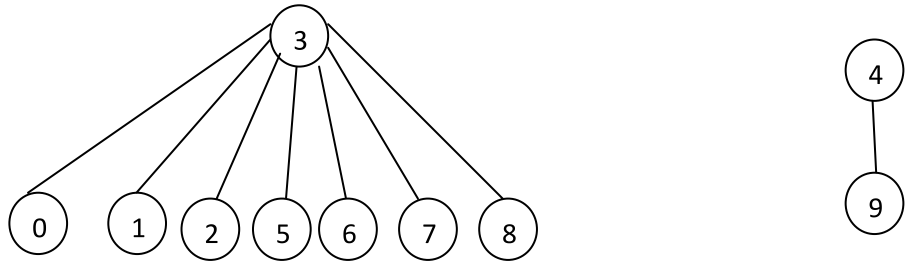
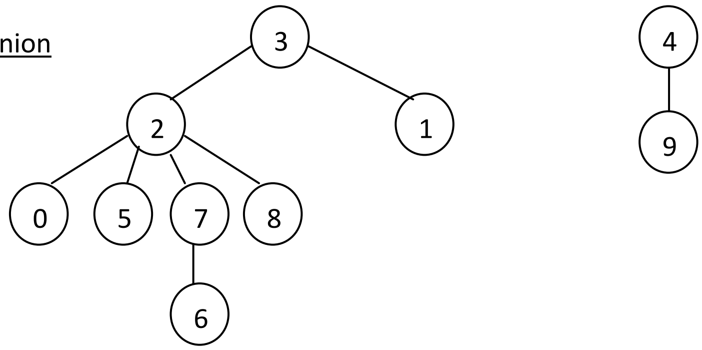
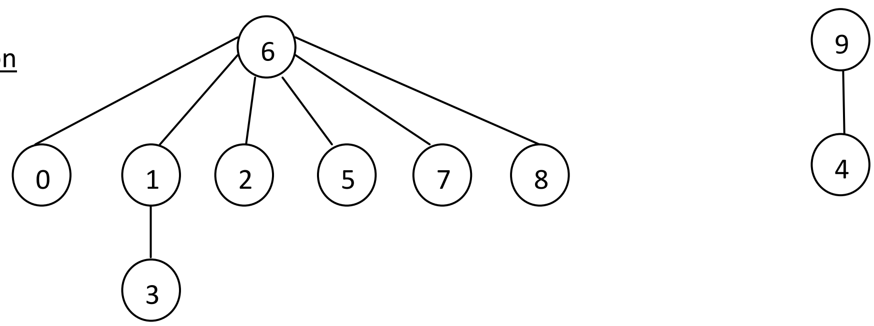
1. a) Quick find



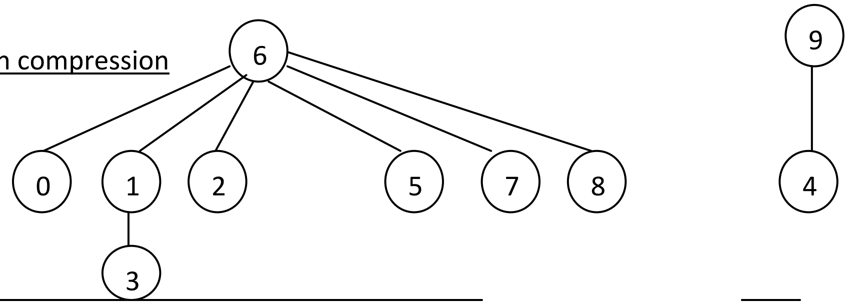
b) Quick Union



c) Weighted Quick Union



d) Weighted Quick Union with path compression

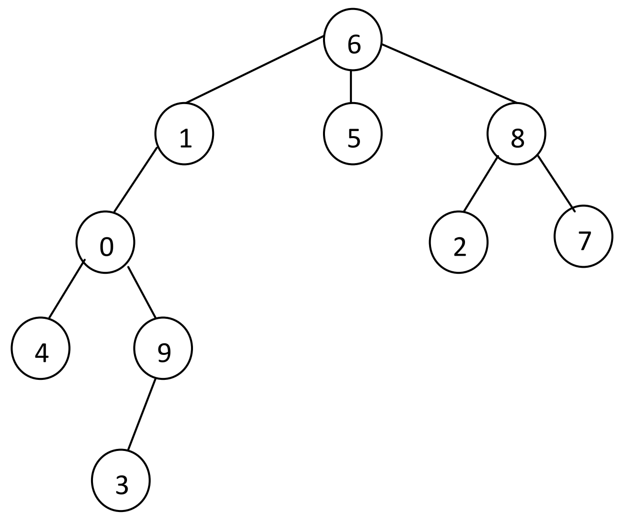


The trees produced in (c) and (d) are identical,if we follow path compression by halving.Otherwise if full path compression is used, then the trees would look like

6 9

0 1 2 3 5 7 8 4

2.



This tree cannot be the result of running weighted quick union without path compression.

The only way node 0 can be put under node 1 is with a union(p,q) operation where p is 1 and q is one of 0,4,9,3 or vice versa.

Let's call this the target union(p,q) operation.

In either case ,(4,9,3) must have been put below 0 before the target union(p,q) operation took place, because otherwise (4,9,3) would directly connect under 1 and not 0.

Also at the time the target union(p,q) operation took place, node 1 was not connected under node 6 because otherwise node 0 would directly connect under node 6.

It is clear that the subtree containing node 0 was bigger than the subtree containing the node 1(at the time the target union(p,q) operation took place).

That means that if the algorithm being used was the weighted quick union, then node1 must have been put before node 0 and not the other way round.

3. a) The outer loop is executed 'n' times and the inner loop is executed 10 times for each execution of the outer loop. So the loop body is executed 10n times. (As another way to think about it the number of times the loop body is executed is given by

$$\sum_{i=0}^{n-1}\sum_{j=1}^{10} 1 = \sum_{i=0}^{n-1} 10 = 10n$$

Now as per the question g(n) = 10n.

i) If g(n) ~ f(n) then $\lim_{n\to\infty}\frac{g(n)}{f(n)} = 1$ and so $\lim_{n\to\infty}\frac{10n}{f(n)} = 1$ thus f(n) = 10n
ii) If g(n) = O(f(n)) then f(n) = n (with c = 20 and $n_0 = 1$ e.g 20n > 10n for n>=1)
iii) If g(n) = Ω(f(n)) then f(n) = n. (with c = 1 and $n_0 = 1$)

b) The first loop has time complexity n . For the pair of nested loops ,the loop body is executed

$$\sum_{i=0}^{n-1}\sum_{j=i}^{n-1} 1 = \sum_{i=0}^{n-1} n - i = (n + (n-1) + \cdots + 1) = \frac{n(n+1)}{2} \text{ times}$$

(The first step above follows since there are ((n – 1) – i + 1 = n – i) values of j for each value of i).

So the second set of loops has overall time complexity $\frac{n(n+1)}{2}$.Thus the overall time complexity of the entire program segment is $\frac{n(n+1)}{2} + n = \frac{n^2+3n}{2}$

Now as per question g(n) =

i)If g(n) ~ f(n) then $\lim_{n\to\infty}\frac{g(n)}{f(n)} = 1$ and so $\lim_{n\to\infty}\frac{n^2+3n}{2.f(n)} = 1$ thus f(n) =$n^2/2$

ii)If g(n) = O(f(n)) then f(n) = n^2.

(e.g c = 2 , n_0 =2 , So, 2$n^2 > \frac{n^2+3n}{2}$ => 1.5$n^2 > 1$ .5n => $n > 1$)

iii)If g(n) = Ω(f(n)) then f(n) = n^2.

(e.g c = 0.5 , n_0 =1 , So, 0.5$n^2 < 0.5n^2 + \frac{3n}{2}$ => 0 < 1.5n)

4.

If $\lim_{n\to\infty}\frac{f(n)}{g(n)} = 0$ then $f(n)$ grows at a slower rate compared to $g(n)$.

If $\lim_{n\to\infty}\frac{f(n)}{g(n)} = c, where\ c\ is\ a\ constant$ then $f(n)$ grows at the same rate compared to $g(n)$.

If $\lim_{n\to\infty}\frac{f(n)}{g(n)} = \infty,$ then $f(n)$ grows at a faster rate compared to $g(n)$.

Since $\lim_{n\to\infty}\frac{n}{2^n} = 0$ => $n.2^{-n} = \Theta(1)$.The rest of the functions for n->∞ gives ∞ . So this is the smallest growing function…………………………(1)

$$\lim_{n\to\infty}\frac{\sqrt{n}}{\frac{n}{(\log n)^4}} = \lim_{n\to\infty}\frac{(\log n)^4}{\sqrt{n}} = \lim_{n\to\infty}\frac{4(\log n)^3.\left(\frac{1}{n}\right).\left(\frac{1}{\ln 2}\right)}{\frac{1}{2\sqrt{n}}}$$

(∞/∞ $form, By\ Using\ L'\ Hospitals\ rule$)

$$= \lim_{n\to\infty}\frac{\left(\frac{1}{\ln 2}\right)8(\log n)^3}{\sqrt{n}}$$ =(∞/∞ $form, By\ Using\ L'\ Hospitals\ rule$)

$$= \lim_{n\to\infty}\frac{\left(\frac{1}{\ln 2}\right)^2 8.3.2(\log n)^2}{\sqrt{n}} = \lim_{n\to\infty}\frac{\left(\frac{1}{\ln 2}\right)^3 8.3.2.2\left(\log n)\left(\frac{1}{n}\right)\right]}{\sqrt{n}}$$

(∞/∞ $form, By\ Using\ L'\ Hospitals\ rule$)

$$= \lim_{n\to\infty}\frac{\left(\frac{1}{\ln 2}\right)^3 96(\log n)]}{\sqrt{n}} = \lim_{n\to\infty}\frac{\left(\frac{1}{\ln 2}\right)^4 .96.\left(\frac{1}{n}\right)}{\frac{1}{2\sqrt{n}}}$$ (∞/∞ $form, By\ Using\ L'\ Hospitals\ rule$)

$$= \lim_{n\to\infty}\frac{192\left(\frac{1}{\ln 2}\right)^4}{\sqrt{n}}$$ = 0

So, in terms of growth rate……………………(2)

$\lim_{n\to\infty} \frac{(\log n)^2}{\sqrt{n}} = \lim_{n\to\infty} \frac{2\left(\log(n)\left(\frac{1}{n}\right)\left(\frac{1}{\ln 2}\right)\right)}{\frac{1}{2\sqrt{n}}}$ (∞/∞ *form, By Using L'Hospitals rule*)

$= \lim_{n\to\infty} \frac{4\left(\log(n)\left(\frac{1}{\ln 2}\right)\right)}{\sqrt{n}} = \lim_{n\to\infty} \frac{4\left(\frac{1}{n}\right)\left(\frac{1}{\ln 2}\right)^2}{\frac{1}{2\sqrt{n}}}$ (∞/∞ *form, By Using L'Hospitals rule*)

$= \lim_{n\to\infty} \frac{8\left(\frac{1}{\ln 2}\right)^2}{\sqrt{n}} = 0$

So, in terms of growth rate……………………(3)

$\lim_{n\to\infty} \frac{\log\log n}{(\log n)^2} = \lim_{n\to\infty} \frac{\left(\frac{1}{\log n}\right)\left(\frac{1}{n \ln 2}\right)}{2\log n \left(\frac{1}{n}\right).\left(\frac{1}{\ln 2}\right)}$ (∞/∞ *form, Using L'Hospitals rule*)

$= \lim_{n\to\infty} \frac{1}{2(\log n)^2} = 0$

So, $\log\log n < (\log n)^2$ in terms of growth rate…………………..(4)

Combining (1) , (2) , (3) , (4) , the order of the functions in increasing Θ order is :

$n.2^{-n} < \log\log n < (\log n)^2 < \quad < \frac{n}{(\log n)^4}$

5. $C_N = C_{N/2} + N^2$; $C_1 = 0$;for $N \geq 2$

k

Let N =

$C_N - C_{N/2} = N^2$

$C_{N/2}$ - $C_{N/4} = \left(\frac{N}{2}\right)^2$

…………………………….

$C_{N/(2^{(k-1)})}$ - $C_{N/(2^k)}$

Adding both sides we get,

C_N - $C_{N/(2^k)} = N^2 [1 + ((1/2)^2 + (1/2)^4 + \cdots + (1/2)^{(2k-2)})]$

C_N - $C_1 = N^2 [1 + ((1/4)^1 + (1/4)^2 + \cdots + (1/4)^{(k-1)})]$

$C_N = N^2 \frac{1-\left(\frac{1}{4}\right)^k}{1-\left(\frac{1}{4}\right)} =$, Since $k = \log_2 N$

Now Let $0 \le C_N \le C$., where C is a constant

So, $C \ge \frac{4}{3\left[1-\left(\frac{1}{4}\right)^{\log_2 N}\right]}$

If we choose $n_0 = 2$ and C = 2 then all conditions of Big oh are satisfied.

So, $C_N = O(N^2)$ and $f(n) = N^2$

6. Suppose $n^4 = O(3n^3)$

Then there would be witnesses n_0 and C such that $n^4 \le C.3n^3$ for all $n \ge p$. (where C and p are positive constants)

i.e Then $n^4 \le Kn^3$ for all $n \ge p$(where K = 3C is a positive constant)

But if we pick n_1 equal to the larger of 2K and p, then the following inequality must hold

$n_1^4 \le K . n_1^3$………<u>equation 1</u> (because $n_1 \ge n_0$ and $(n_1^4) \le Kn^3$ allegedly holds for all $n \ge n_0$)

Now if we divide both sides of <u>equation 1</u> by n_1 ^ 3 then we have n_1,but we also choose n_1 to be at least 2K. Since K must be positive (as C must be positive), n_1 cannot be both less than K and greater than 2K.

Thus $n^4 = O(3n^3)$ is not true.