CSCI 260 Test: Big-O, Ω, Θ Notation and Union-Find Forests A

Oct 6, 2022 Instructor: Gara Pruesse

Out of 37 marks

if you don't have A here, see B.

1. (2 marks) Prove the following claim using only the definition of Big-O and valid transformations around the less-than-or-equal-to relation:

Claim: $4n^2 - n + 16 \in O(n^2)$

$$4n^2 - n + 16 \le 4n^2 + 16 \quad \forall n \ge 1$$
$$\le 5n^2 \quad \forall n \ge 4$$

$\therefore 4n^2 - n + 16 \in O(n^2)$, as certified by $n = 4$, $c = 5$.

2. (4 marks) Prove the following Claim.

Claim: $n^4 - n$ is **not** in $O(n^3)$

Proof: BWOC. Suppose $n^4 - n \in O(n^3)$. Then by def^n big Oh, $\exists$ constants $c > 0$, $n_0 > 0$ s.t. $n^4 - n \le c \cdot n^3 \; \forall n \ge n_0$

$\Rightarrow n^4 \le cn^3 + n \quad \forall n \ge n_0$

$\Rightarrow n^4 \le cn^3 + n^3 \quad \forall n \ge \max(n_0, 1)$

$\Rightarrow n \le c + 1 \quad \forall n \ge \max(n_0, 1)$.

$\Rightarrow \Leftarrow \quad \therefore n^4 - n \notin O(n^3)$

3. (4 marks) Prove the following claim using the Rules (Facts) of Big-Oh. Do not use the less-than rule.

Claim: $\frac{1}{3}n\log n + \log n \ \epsilon\ O(n^2 \log n)$

1. $1 \in O(n^2)$ Polynomial
2. $\log n \in O(\log n)$ CF
3. $\log n \in O(n^2 \log n)$ 1,2 Product
4. $\frac{1}{3}n \in O(n^2)$ Polynomial
5. $\frac{1}{3}n\log n \in O(n^2 \log n)$ 2,4 Product
6. $\frac{1}{3}n\log n + \log n \in O(n^2 \log n)$ 3,5 Sum Rule. ▨

4. (4 marks) Prove using either method the following claim. Do not use the less than rule.

Can use Theorem: $1 \in O(\log n)$.

Claim: $(4n^2 + 7n)/\log n \ \epsilon\ O(n^3)$

1. $1 \in O(\log n)$ Theorem
2. $\frac{1}{\log n} \in O(1)$ 1, Recip
3. $4n^2 + 7n \in O(n^3)$ Polynomial
4. $(4n^2+7n)/\log n \in O(n^3)$ 2, 3 Product Rule. ▨

5. (4 marks) Prove the following claim using the Rules (Facts) of Big-Oh. Do not use the less-than rule.

Claim: $5n^3 \ \epsilon\ O(n^4 \log n)$

1. $1 \in O(\log n)$ Theorem.
2. $5n^3 \in O(n^4)$ Polynomial Rule
3. $5n^3 \in O(n^4 \log n)$ 1,2 Product Rule. ▨

6. (6 marks) Use the Master Theorem to determine the running time if the recurrence relation for the running time is the following:

$n^{\log_b a} = n^{\log_5 3} \cong .68$

a) T(n) = 3T(n/5) + n^2

$n^2 \in \Omega(n^{.68+\varepsilon})$

and $\exists c$, $0<c<1$ such that $3\left(\frac{n}{5}\right)^2 < c \cdot n^2$

ie such that $\frac{3}{25} n^2 < cn^2$. Eg $c = \frac{4}{25}$ suffices.

$\therefore$ by MT case 3, $T(n) \in \Theta(n^2)$

b) T(n) = 5T(n/3) + n^2 $n^{\log_b a} = n^{\log_3 5} = n^{1.47}$

$n^2 \in \Omega(n^{1.47+\varepsilon})$

and $\exists c$, $0<c<1$ s.t $5\left(\frac{n}{3}\right)^2 < cn^2$

ie such that $\frac{5}{9} n^2 < cn^2$; e.g. $c = \frac{6}{9}$ will suffice

$\therefore$ by MT case 3, $T(n) \in \Theta(n^2$

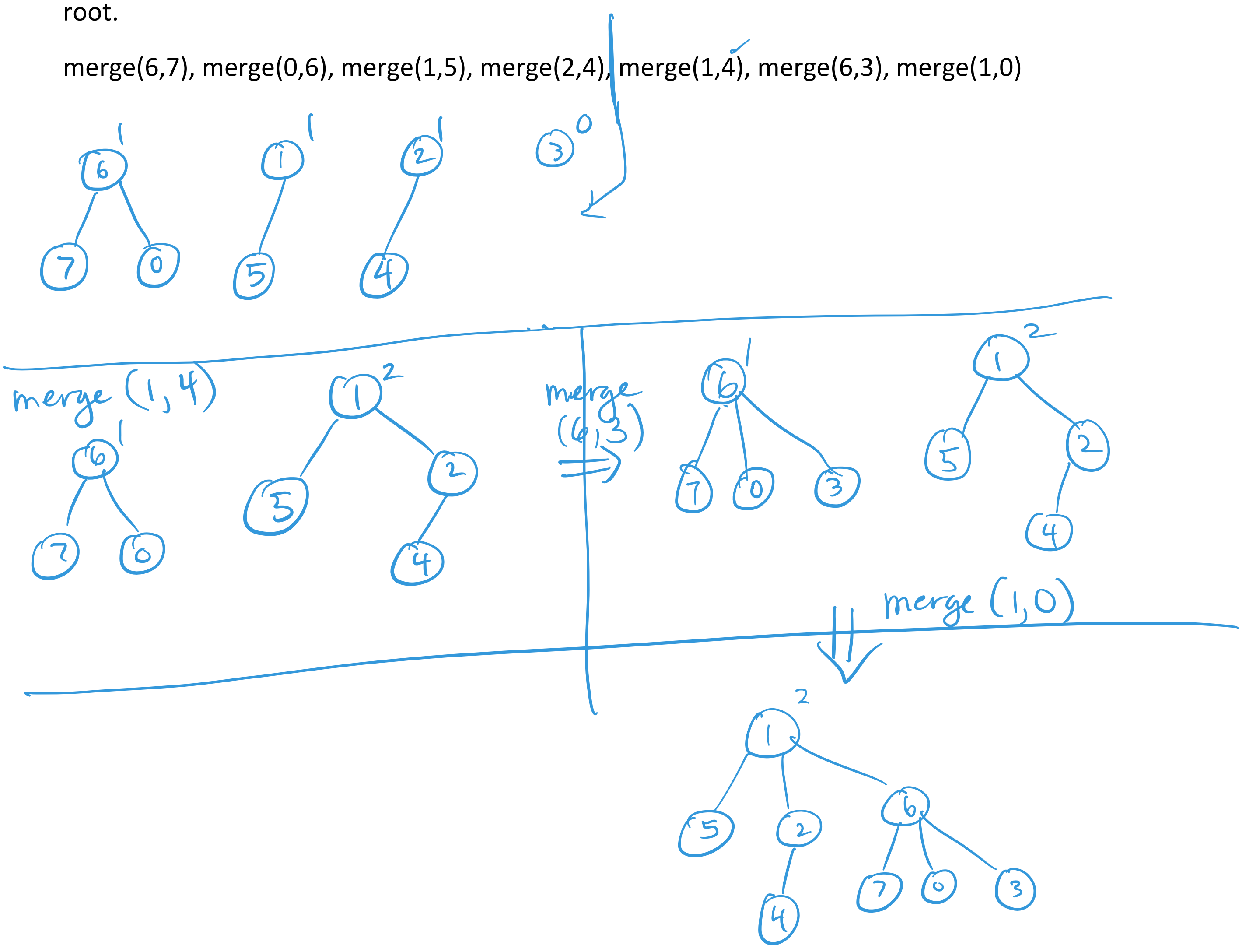
7. (4 marks) Sort the following functions in terms of their rate of growth, in the Big-Oh sense. Put the slowest growing functions on the left (the slowest growing functions are running time of the fastest algorithms). If two functions are in the Theta of one another, i.e., in the Big-Oh sense they have the same running time, then group them together in curly braces.

$n^2 \log n$ $\quad n^{1.5}$ $\quad \overset{n}{\log(2^n)}$ $\quad 2^{2n}$ $\quad \overbrace{2^{\log n + \log\log n}}^{n \log n}$ $\quad n^2\sqrt{\log n}$ $\quad 150n^2$ $\quad n^{1.5}\log n$

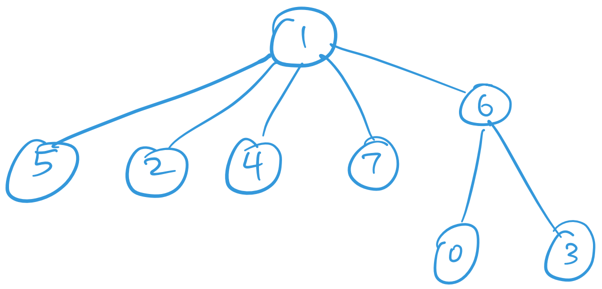
$\log(2^n)$, $2^{\log n + \log\log n}$, $n^{1.5}$, $n^{1.5}\log n$, $150n^2$, $n^2\sqrt{\log n}$, $n^2 \log n$, 2^{2n}

8. a) (5 marks) Assume a disjoint set starts with 8 elements, each named by a number from 0 to 7, each in their own set, upon which the following sequence of merge operations are performed, one after another. Show all stages of the forest as it evolves under the operations, assuming the **Quick-Union implementation** of union-find, where union-by-rank **is** implemented but path compression **is no**t. If two trees have the same rank, then let the first parameter tree be the one that is made into the root of the resulting tree. Sthe results just by drawing the tree that results (you don't have to redraw, at each stage, the trees that do not change in that stage); be sure to indicate the rank of the root.

merge(6,7), merge(0,6), merge(1,5), merge(2,4), merge(1,4), merge(6,3), merge(1,0)



b) (2 marks) Show the resulting tree if, after that series of merges, the find operation now has path-compression, and merge(7,4) is called. You need only show the affected tree. Show the rank of the root.



10. (2 marks) Give the recurrence relation that is the running time of the following procedure.

```
Shuffle( A[1..n]) {
        Shuffle( A[1..n/4] )
        Shuffle( A[2n/4.. 3n/4])
        Shuffle( A[n/4..2n/4)
        Do $\Theta(n^{1.5})$ work here
}
```

$$T(n) = 3T\left(\frac{n}{4}\right) + n^{1.5}$$