

CSCI 260 Test: Big-Oh Notation and Union-Find Forests

Oct 6 2022 Instructor: Gara Pruesse

Out of 37 marks

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

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

$$\begin{aligned} 4n^2 - n + 16 &\leq 4n^2 + 16 \quad \forall n \geq 1 \\ &\leq 5n^2 \quad \forall n \geq 4 \end{aligned}$$

◦ $4n^2 - n + 16 \in O(n^2)$, demonstrated by
 $c=5, n_0=4$.

2. (4 marks) Prove using either method the following claim. Do not use the less-than rule.

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

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

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 Rule
4. $(4n^2 + 7n)/\log n \in O(n^3)$ 2, 3 Product Rule. \square

3. (6 marks) Use the Master Theorem to determine the running time if the recurrence relation for the running time is the following:

a) $T(n) = 3T(n/5) + n^2$ $n^{\log_b a} = n^{\log_5 3} \approx n^{0.68}$

$n^2 \in \Omega(n^{0.68 + \epsilon})$; and $\exists c, 0 < c < 1$, such that

$3(\frac{n}{5})^2 < cn^2$, i.e. such that $\frac{3}{25}n^2 < n^2$,

for example $c = \frac{4}{25}$ will work.

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

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

$n^2 \in \Omega(n^{1.47 + \epsilon})$, and $\exists c, 0 < c < 1$, such that
 $5(\frac{n}{3})^2 < cn^2$ i.e. $\frac{5}{9} < c$; for example, $c = \frac{6}{9}$ will work.

∴ By MT case 3, $T(n) \in \Theta(n^2)$

4. (4 marks) Prove the following Claim.

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

Proof: BWOC. Suppose $n^4 - n \in O(n^3)$

$\Rightarrow \exists$ ^{constants} $c > 0, n_0 \geq 1$ such that $n^4 - n \leq c \cdot n^3 \forall n \geq n_0$.

$\Rightarrow n^4 \leq c \cdot n^3 + n \forall n \geq n_0$

$\Rightarrow n^4 \leq c \cdot n^3 + n^3 \forall n \geq \max(n_0, 1)$

$\Rightarrow n \leq c + 1 \forall n \geq \max(n_0, 1)$

$\Rightarrow \Leftarrow$

∴ $n^4 - n \notin O(n^3)$ ◻

5. (4 marks) Prove the following claim using the Rules (Facts) of Big-O.

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

1. $\log n \in O(n)$ SBT Log Dom Rule
2. $\frac{1}{n} \in O(\frac{1}{\log n})$ 1, Recip.
3. $5n^4 \in O(n^4)$ CF
4. $5n^3 \in O(n^4 / \log n)$ 2,3 Product Rule. \square

6. (4 marks) Prove the following claim using the Rules (Facts) of Big-O.

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

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 Rule.
6. $\frac{1}{3}n \log n + \log n \in O(n^2 \log n)$ 3,5, Sum Rule \square

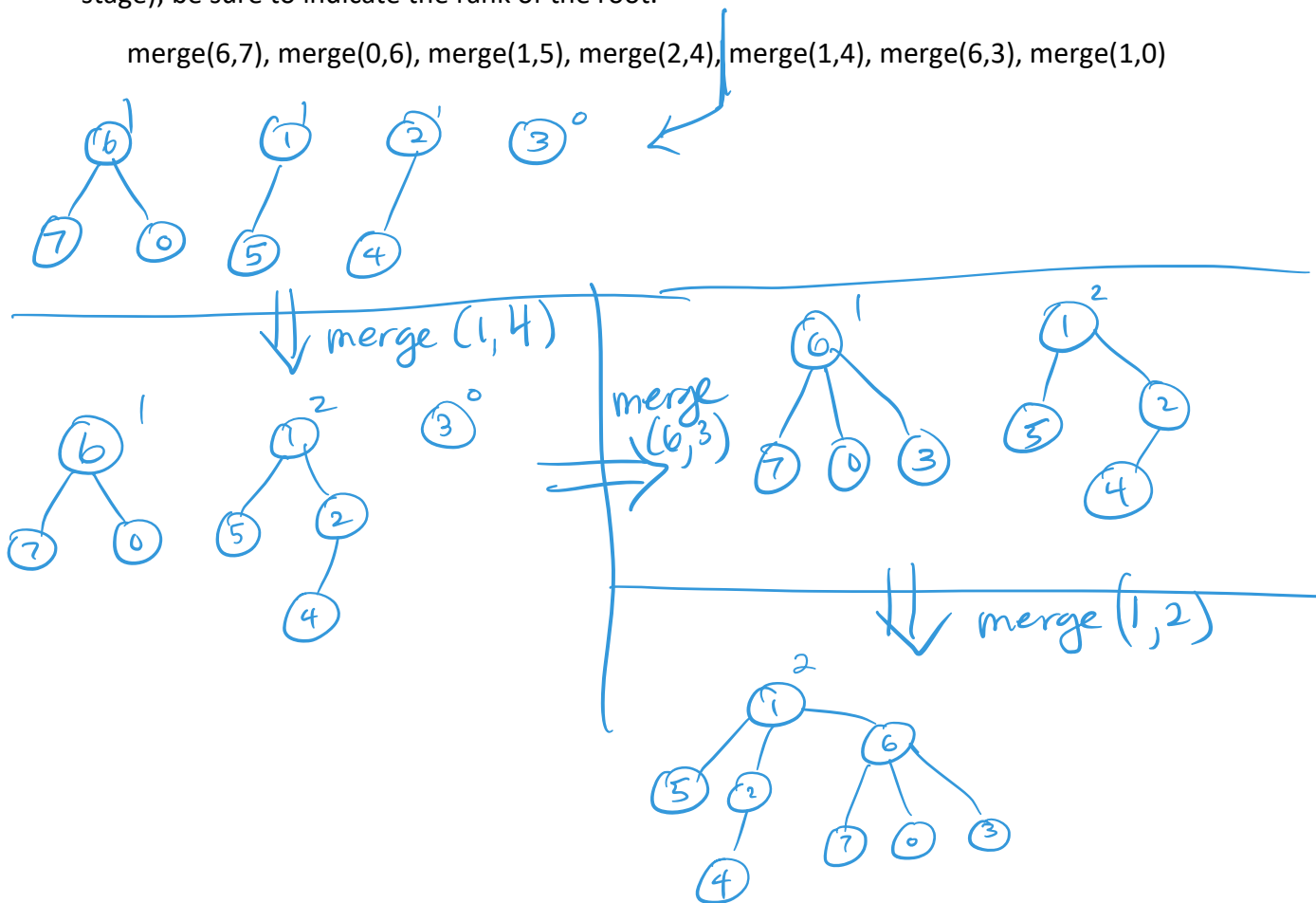
7. (4 marks) Sort the following functions in terms of their rate of growth, in the Big-O sense. Put the slowest growing functions on the left (the slowest growing functions are running time of the fastest algorithms). If two functions are in the Θ of one another, i.e., in the Big-O sense they have the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

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

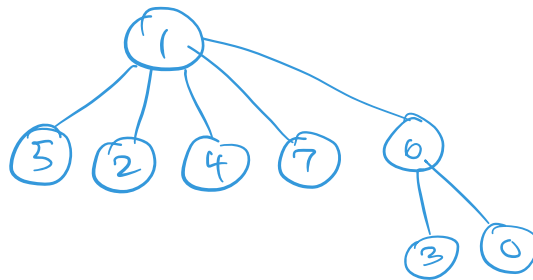
$\log 2^n$, $2^{\log n + \log \log n}$, $n^{1.5}$, $n^{1.5} \log n$, $150n^2$, $n^2 \sqrt{\log n}$, $n^2 \log n$, 2^{2n}

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 **not**. 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. You may give the forests as a series of values of the parent array data structure (be sure to also include the rank array data structure); or you may show the 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.



9. (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  
}
```

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