CS 3130 **Design and Analysis of Algorithms** **Fall 2018**

KEY TO HOMEWORK #2 [40 points].

Due date: Wednesday, October 7.
(Homework will be collected before the lecture!)

A. **Read Chapter 3 and do the following exercises**:

1. ***Explain***, if 3^{n+1} is: $O(3^n)$; $\Omega(3^n)$; $\Theta(3^n)$.

Solution.

Since $\lim_{n\to\infty}\frac{3^{n+1}}{3^n}=3\neq 0$ and 3 is not infinity, 3^{n+1} is $\Theta(3^n)$. According to the definition (p.44), there exist c_1, c_2 and n_0 such that $c_1\cdot 3^n \le 3^{n+1} \le c_2\cdot 3^n$ for all $n>n_0$. From the right side of this inequality: $3^{n+1}\le c_2\cdot 3^n$ for all $n>n_0$, which means that 3^{n+1} is $O(3^n)$. From the left side of this inequality: $c_1\cdot 3^n \le 3^{n+1}$ for all $n>n_0$, which means that 3^{n+1} is $\Omega(3^n)$.

2. ***Explain***, if 3^{3n} is: $O(3^n)$; $\Omega(3^n)$; $\Theta(3^n)$.

Solution.

$\lim_{n\to\infty}\frac{3^{3n}}{3^n}=\lim_{n\to\infty}3^{2n}=\infty$, 3^{3n} is $\omega(3^n)$ and, correspondingly, $\Omega(3^n)$. However, since 3^{3n} is $\omega(3^n)$, then 3^{3n} can't be $O(3^n)$ and, consequently, it is not $\Theta(3^n)$.

3. Fill in the table below by indicating whether the function $f(n)$ is O, o, Ω, ω, or Θ of $g(n)$. Your answer should be the table with either "yes" or "no" in each cell.
<u>Answers without proof will not count!</u>

$f(n)$	$g(n)$	O	o	Ω	ω	Θ
8^n	$8^{n/2}$	No	No	Yes	Yes	No
$\sqrt{n^k}$ (k is an even positive integer)	2^n	Yes	Yes	No	No	No
$\lg(n^n)$	$\lg(n!)$	Yes	No	Yes	No	Yes

Solution.

(a) $\lim_{n\to\infty}\frac{f(n)}{g(n)}=\lim_{n\to\infty}\frac{8^n}{8^{n/2}}=\lim_{n\to\infty}8^{n/2}=\infty$. Then 8^n is $\omega(8^{n/2})$ and, consequently, $\Omega(8^{n/2})$. Then 8^n can't be $o(8^{n/2})$, $O(8^{n/2})$ and $\Theta(8^{n/2})$.

(b) $\lim\limits_{n\to\infty}\dfrac{f(n)}{g(n)}=\lim\limits_{n\to\infty}\dfrac{n^{k/2}}{2^n}=\lim\limits_{x\to\infty}\dfrac{x^{k/2}}{2^x}=\lim\limits_{x\to\infty}\dfrac{\frac{k}{2}\cdot x^{(k/2)-1}}{2^x(\ln 2)}=\lim\limits_{x\to\infty}\dfrac{\frac{k}{2}\cdot\left(\frac{k}{2}-1\right)x^{(k/2)-2}}{2^x(\ln 2)^2}=\ldots=\lim\limits_{x\to\infty}\dfrac{\frac{k}{2}\cdot\left(\frac{k}{2}-1\right)\cdot\left(\frac{k}{2}-2\right)\cdot\ldots\cdot 1}{2^x(\ln 2)^{k/2}}=0 \Rightarrow \sqrt{n^k}$ is $o(2^n) \Rightarrow \sqrt{n^k}$ is $O(2^n)$. Since $\sqrt{n^k}$ is $o(2^n)$, then $\sqrt{n^k}$ can't be $\omega(2^n)$ and $\Omega(2^n)$. So, $\sqrt{n^k}$ is not $\Theta(2^n)$.

(c) $\lim\limits_{n\to\infty}\dfrac{f(n)}{g(n)}=\lim\limits_{n\to\infty}\dfrac{\lg(n^n)}{\lg(n!)}=1$ (see exercise 3.2-3 (a), p.60). $\Rightarrow \lg(n^n)$ is $\Theta(\lg(n!))$ and, consequently, $O(\lg(n!))$, and $\Omega(\lg(n!))$. Also, $\lg(n^n)$ is not $o(\lg(n!))$ and $\omega(\lg(n!))$.

4. Problem 3 – 3 (a), pp.61 – 62.
<u>NOTE:</u> (1) from 30 functions listed in the book, do this exercise only for the functions, positions of which are marked by 'X' in the table below:

		X	X	X	
X	X	X	X		
X		X		X	X
X		X	X	X	X
		X	X	X	

(2) Please, rank exponential functions according to their bases.
(3) Partition your list into equivalence classes such that functions $f(n)$ and $g(n)$ are in the same class if and only if $f(n)\in\Theta(g(n))$.
(4) Explanations are not required.

<u>Solution.</u>
In the order of increasing growth (functions from the same equivalence class are enclosed into braces):

$1;\ \ln(\ln n);\ \sqrt{\lg n};\ \ln n;\ \lg^2 n;\ \left(\sqrt{2}\right)^{\lg n}=\sqrt{n};\ \{n;\ 2^{\lg n}=n\};\ \{n\lg n;\ \lg(n!)\};$

$\{n^2; 4^{\lg n}=n^2\};\ n^3;\ \left(\dfrac{3}{2}\right)^n;\ 2^n;\ n\cdot 2^n;\ e^n;\ n!;\ (n+1)!$

B. **Read Chapter 4 and handout on recursion. Do the following exercises**:
5. Use the method of backward substitution (telescoping) to solve the recurrence:
$T(n)=2T(n-1)+1;\ T(1)=5$.

<u>Solution.</u>
$T(n)=2T(n-1)+1=2[2T(n-2)+1]+1=2^2T(n-2)+2+1=2^2[2T(n-3)+1]+2+1=$

$=2^3T(n-3)+2^2+2+1=\ldots=2^{n-1}T(1)+2^{n-2}+2^{n-3}+\ldots+2+1=2^{n-1}\cdot 5+1\cdot\dfrac{2^{n-1}-1}{2-1}=$

$= 2^{n-1} \cdot 5 + 2^{n-1} - 1 = 2^{n-1} \cdot 5 + 2^{n-1} - 1 = 6 \cdot 2^{n-1} - 1$

6. For the recurrence $T(n) = \begin{cases} 1 \ \textit{if} \ n = 1 \\ 3T\left(\frac{n}{3}\right) + 1 \ \textit{if} \ n > 1 \end{cases}$

   show that $T(n) \in \Theta(n)$. (Make sure that you prove two facts: $T(n) \in O(n)$ and $T(n) \in \Omega(n)$). Consider only $n = 3^k$. *Hint*: use the idea of subtracting the lower-power term (see p.85).

Solution.

(a) To prove that $T(n) \in O(n)$, we will show that there exist C>0 and b such that $T(n) < Cn - b$ (*) for sufficiently big values of n.

Assume that (*) is true for $\frac{n}{3}$, which means that $T\left(\frac{n}{3}\right) \in O\left(\frac{n}{3}\right)$, or $T\left(\frac{n}{3}\right) < C\frac{n}{3} - b$. From the recurrence:

$T(n) = 3T\left(\frac{n}{3}\right) + 1 < 3\left(C\frac{n}{3} - b\right) + 1 = Cn - 3b + 1$. The inequality $T(n) < Cn - b$ will be true when $-2b + 1 < 0$, or $b > \frac{1}{2}$.

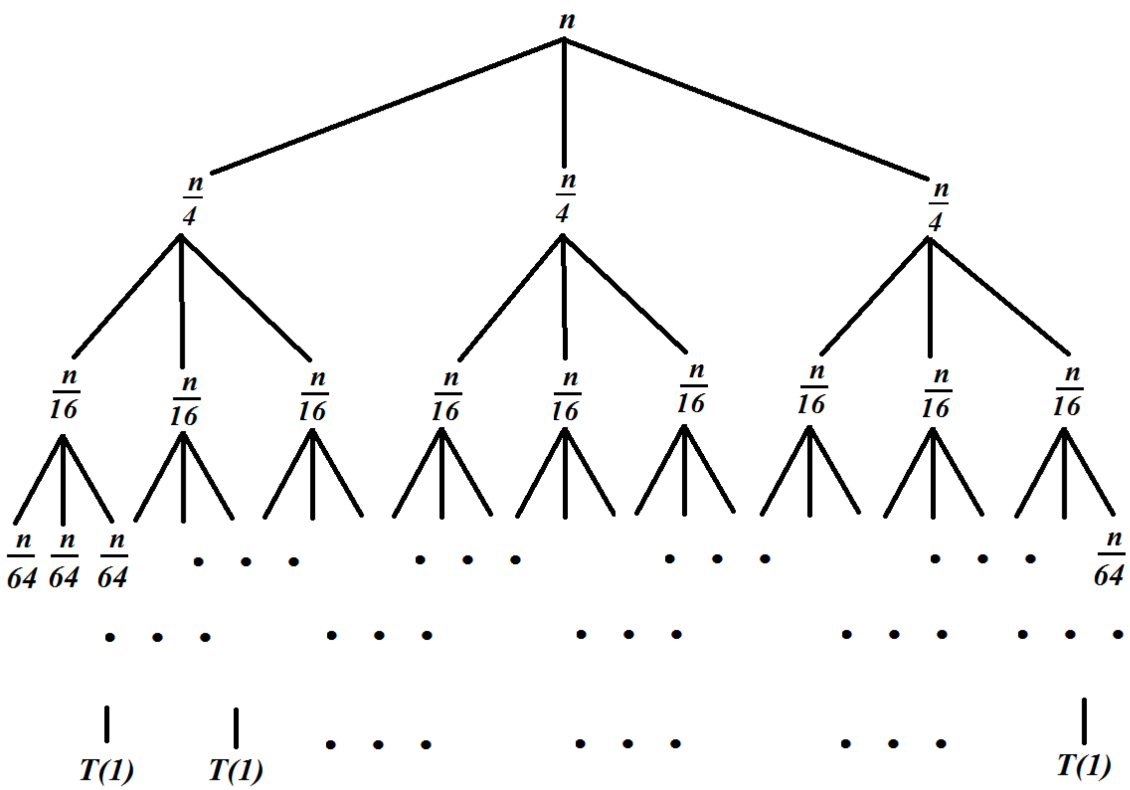
(Note that if the condition (*) does not have a constant term –*b*, the proof is not correct.)

(b) In this part, we should show that that there exist D>0 and k such that $T(n) > Dn - k$ for sufficiently big values of n. After the similar assumption and calculations, we get: $T(n) = 3T\left(\frac{n}{3}\right) + 1 > 3\left(D\frac{n}{3} - k\right) + 1 = D - 3k + 1$. The inequality $T(n) > Dn - k$ will be true when $-2k + 1 > 0$, or $k < \frac{1}{2}$.

7. Use the recursion tree method to find a big-Θ estimate for the solution of the recurrence $T(n) = 3T\left(\frac{n}{4}\right) + n$.

Solution.

The given recurrence corresponds to the recursion tree shown below.



Assume that $T(1) = A$ and summarize execution times at all levels:

$$T(n) = n + \frac{3}{4}n + \frac{3^2}{4^2}n + \frac{3^3}{4^3}n + ... + \frac{3^{h-1}}{4^{h-1}}n + L \cdot T(1) < n\sum_{k=0}^{\infty}\left(\frac{3}{4}\right)^k + A \cdot L = \frac{n}{1-\frac{3}{4}} + An^{\log_4 3} =$$

$= 4n + An^{\log_4 3}$. Since $\log_4 3 < 1$, $T(n)$ has the same order of growth as the term $4n$ $\Rightarrow$ $T(n) \in \Theta(n)$.

8. Exercise 4.5 – 1, p.96.

Solution.

For all parts of this problem: a=2, b=4. Only a parameter d will have different values:

(a) $d = 0$ and a=2 > $b^d = 4^0 = 1$. So, $T(n) \in \Theta(n^{\log_b a})$ or $T(n) \in \Theta(n^{\log_4 2})$	(b) $d = 0.5$ and a=2 = $b^d = \sqrt{4} = 2$. So, $T(n) \in \Theta(n^d \log n)$ or $T(n) \in \Theta(\sqrt{n}\log n)$
(c) $d = 1$ and a=2 < $b^d = 4^1 = 4$. So, $T(n) \in \Theta(n^d)$ or $T(n) \in \Theta(n)$	(d) $d = 2$ and a=2 < $b^d = 4^2 = 16$. So, $T(n) \in \Theta(n^d)$ or $T(n) \in \Theta(n^2)$

9. Answer the following questions (justify your answers!):

(a) For the given version of a QUICKSORT algorithm, are arrays made up of all equal elements the worst-case input, the best-case input or neither?

(b) For the given version of a QUICKSORT algorithm, are strictly decreasing arrays the worst-case input, the best-case input or neither?

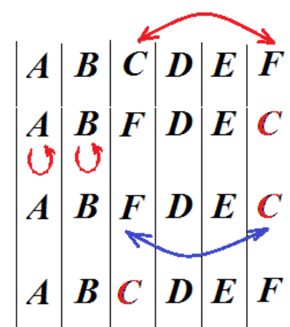
Answer for both (a) and (b):

Arrays of both types are the worst-case input, since each call to the function PARTITION produces one subarray of size n-1, and one subarray with 0 elements.

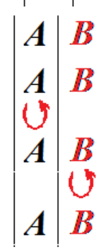
10. (a) Show exactly the progress of the QUICKSORT algorithm with the median-of-3 partition described in the textbook for the array: <A, B, C, D, E, F>. Explain if strictly increasing arrays are the worst-case input, the best-case input or neither for the given version of a QUICKSORT algorithm with the median-of-3 partition.

Solution.

(For both parts of this question, the pivot element is shown in red.) During the first call to the PARTITION function, the pivot element is chosen as the median of the array, which is C. So, the rest of the function will work with the array: <A, B, F, D, E, $\boxed{C}$>. The result of the first call to the PARTITION is shown on the right.



After that, the the median-of-3 approach is not applicable to the left subarray since it has only 2 elements, so we may call a basic form of the function PARTITION for this subarray. The progress of PARTITION is shown on the right.

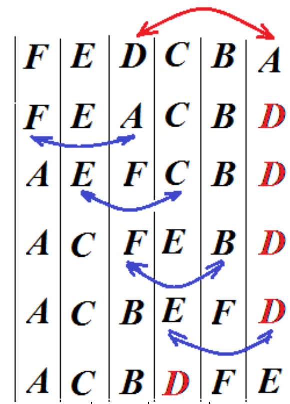


In the right subarray, the median element is E, it will be exchanged with F and become a pivot. The progress of the PARTITION function is shown on the right. After that, the work of QUICKSORT is completed.

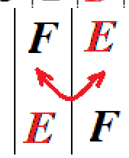
(b) Show exactly the progress of the QUICKSORT algorithm with the median-of-3 partition described in the textbook for the array: <F, E, D, C, B, A>. Explain if strictly decreasing arrays are the worst-case input, the best-case input or neither for the given version of a QUICKSORT algorithm with the median-of-3 partition.

<u>Solution.</u>

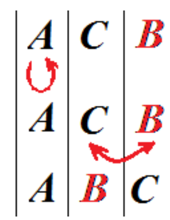
During the first call to the PARTITION function, the pivot element is chosen as the median of the array, which is D. So, the rest of the function will work with the array: <F, E, A, C, B, D>. The result of the first call to the PARTITION is shown on the right.



After that, the the median-of-3 approach is not applicable to the right subarray since it has only 2 elements, so we may call a basic form of the function PARTITION for this subarray. The only action during the call to the PARTITION will be an exchange of these elements.



In the left subarray, the median element is B, and it will become a pivot. The progress of the PARTITION function is shown on the right. After that, the work of QUICKSORT is completed.



<u>*Answer for both (a) and (b)*</u>:

Arrays of both types are the best-case input since each call to the function PARTITION produces two subarrays with the sizes that either are equal or differ by 1.

<u>Problem for 3 extra points.</u>

There is one fake coin among 65 identically looking coins (the fake coin is lighter). In a balance scale, you can compare any two sets of coins. Develop the algorithm to determine a fake coin using not more than 4 comparisons.

<u>*Solution to this problem is posted separately.*</u>