Data Structures and Algorithms

COSC 336 Assignment 2 - Solutions

Exercise 1.

a Find a Θ evaluation for the function $(4n+1)4^{\log(n)}$. (Hint: $4^{\log(n)}$ can be written in a simpler way.)

b Give an example of two functions $t_1(n)$ and $t_2(n)$ that satisfy the relations: $t_1(n) = \Theta(n^2)$, $t_2(n) = \Theta(n^2)$ and $t_1(n) - t_2(n) = o(n^2)$.

c Give an example of a function $t_3(n)$ such that $t_3(n) = \Theta(t_3(2n))$.

d Give an example of a function $t_4(n)$ such that $t_4(n) = o(t_4(2n))$.

(Note: For (b), (c), (d), the functions t_1, t_2, t_3, t_4 you pick must be selected from the common functions we have discussed, namely polynomials, logarithms, exponentials, factorial.)

Solution: (a) We re-write $4^{\log n} = (2^2)^{\log n} = 2^{2\log n} = 2^{\log(n^2)} = n^2$. So, the expression is equal to $(4n+1)n^2$, which is a polynomial of degree 3, so it is $\Theta(n^3)$.

(b) One example is $t_1(n) = n^2 + n, t_2(n) = n^2$. These two fumctions are both $\Theta(n^2)$ but the difference $t_1(n) - t_2(n) = n$ which is $o(n^2)$.

(c) For instance, we can take $t_3(n) = n$. Then $t_3(2n) = 2n = \Theta(n)$, and so $t_3(n) = (1/2)t_3(2n)$, and therefore $t_3(n) = \Theta(t_3(2n))$.

(d) For instance, we can take $t_4(n) = 2^n$. Then $t_4(2n) = 2^{2n}$, and this works because $2^n = o(2^{2n})$. $\square$

Exercise 2. Fill the table from Exercise 3-2, page 61 (3-rd edition) in the textbook (also attached below), except row c, as asked in the exercise. For example the entry on the first cell in the top row is "yes" because $\log^k n = O(n^\epsilon)$. (Note: in row c all the entries are "no", because $n^{\sin n}$ oscillates.)

Solution:

For (a), we use that (log to any power) is strictly less than any polynomial.

For (b), we use that a polynomial is strictly less than any exponential.

For (d), we take into account that $2^n = 2^{n/2} \cdot 2^{n/2}$.

For (e): $n^{\log c} = 2^{\log(n^{\log c})} = 2^{\log c \cdot \log n}$ and $c^{\log n} = 2^{\log(c^{\log n})} = 2^{\log n \cdot \log c}$, and therefore the two functions are exactly equal.

For (f): We have seen in class that $\log(n!) = \Theta(n \log n)$. $\log(n^n) = n \log n$. Therefore the 2 functions are Θ of each other.

$\square$

3-2 Relative asymptotic growths

Indicate, for each pair of expressions (A, B) in the table below, whether A is O, o, Ω, ω, or Θ of B. Assume that $k \geq 1$, $\epsilon > 0$, and $c > 1$ are constants. Your answer should be in the form of the table with "yes" or "no" written in each box.

	A	B	O	o	Ω	ω	Θ
a.	$\lg^k n$	n^ϵ	yes	yes	no	no	no
b.	n^k	c^n	yes	yes	no	no	no
c.	$\sqrt{n}$	$n^{\sin n}$					
d.	2^n	$2^{n/2}$	no	no	yes	yes	no
e.	$n^{\lg c}$	$c^{\lg n}$	yes	no	yes	no	yes
f.	$\lg(n!)$	$\lg(n^n)$	yes	no	yes	no	yes

Exercise 3. For each of the following program fragments give a $\Theta(\cdot)$ estimation of the running time as a function of n.

(a)
```
sum = 0;
for (int i = 0; i< n * n; i++) {
      for(int j =0;  j < n/2; j++)
        sum++;
}
```

Solution: $t(n) = \Theta(n^3)$. Explanation: Inner loop is $\Theta(n)$ and the outer loop is $\Theta(n^2)$. □

(b)
```
 sum = 0;
 for (int i = 0; i< n; i++) {
sum++;}

for(int j = 0;  j < n/2; j++){
      sum++;}
```

Solution: $t(n) = \Theta(n)$. Explanation: there are 2 sequential loops (not nested), each one having runtime $\Theta(n)$. □

(c)
```
 sum = 0;
 for (int i = 0; i< n * n; i++) {
      for(int j = 0;  j < n * n; j++)
           sum++
}
```

Solution: $t(n) = \Theta(n^4)$. Explanation: Inner loop is $\Theta(n^2)$ and the outer loop is $\Theta(n^2)$. □

(d)
```
 sum = 0;
 for (int i = 1; i< n; i = 2*i)
           sum++
```

Solution: $t(n) = \log n$. Explanation: i grows from 1 to n doubling at each iteration. □

(e)
```
 sum = 0;
 for (int i = 0; i< n; i++) {
      for(int j = 1;  j < n * n; j = 2*j)
           sum++
}
```

Solution: $t(n) = n \log n$. Explanation: The inner loop has runtime $\log(n^2) = 2 \log n = \Theta(\log n)$. The outer loop has n iterations. $\square$

Exercise 4. (a) Compute the sum $S_1 = 500 + 501 + 502 + 503 + \ldots + 999$ (the sum of all integers from 500 to 999). Do not use a program.

Solution: This is an arithmetic series with 500 terms, increment $d = 1$ and initial value $a = 500$. Using the formula we obtain $S = 500 \times 500 + (499 \times 500)/2 = 374750$. $\square$

(b) Compute the sum $S_2 = 1 + 3 + 5 + \ldots + 999$ (the sum of all odd integers from 1 to 999). Do not use a program.

Solution:

This is an arithmetic series with 500 terms, increment $d = 2$ and initial value $a = 1$. Using the formula we obtain $S = 250000$. $\square$

(c) A group of 30 persons need to form a committee of 4 persons. How many such committees are possible?

Solution: $\binom{30}{4} = \frac{30 \cdot 29 \cdot 28 \cdot 27}{1 \cdot 2 \cdot 3 \cdot 4} = 27405$. $\square$

(d) Let C_n be the number of committees of 4 persons selected from a group of n persons. Is the estimation $C_n = o(n^3)$ correct? Justify your answer. (Hint: using the formula $\binom{n}{k}$, you can express the number of committees as a function of n.)

Solution: The estimation is not correct. $\binom{n}{4} = \frac{n(n-1)(n-2)(n-3)}{4!}$. This is a polynomial of degree 4, so it is not $o(n^3)$. $\square$

Exercise 5. Find a $\Theta(\cdot)$ evaluation for the sum

$$S = 1^2\sqrt{1} + 2^2\sqrt{2} + 3^2\sqrt{3} + \ldots + n^2\sqrt{n}.$$

In other words, find a function f such that $S = \Theta(f(n))$.

Show the work for both the upper bound and the lower bound. You can use the technique with integrals, or the method with bounding the terms of the sum.

Solution: (a) Using integrals:

The relevant function is $f(x) = x^2\sqrt{x} = x^{5/2}$, which is an increasing function. So

$$\int_0^n f(x)dx \leq S \leq \int_1^{n+1} f(x)dx$$

Using calculus

$$\int_0^n f(x)dx = n^{7/2}/(7/2),$$

and

$$\int_1^{n+1} f(x)dx = (n+1)^{7/2}/(7/2) - 1^{7/2}/(7/2).$$

It follows that $S = \Theta(n^{7/2})$.

Using the "cut-and-bound" technique:

$$S \le n \cdot n^2\sqrt{n} = n^{7/2}.$$

And

$$S \ge (n/2) \cdot (n/2)^2\sqrt{n/2}.$$

So $S = \Theta(n^{7/2})$. $\square$