

Assignment 1 Cover Page

Due: Friday, January 23 at 12:00 noon

- You may choose to work in a group of two, and submit a single assignment report.
- Please read “How to Prepare Assignment Reports” from the course web page.
- Please make your answers clear and succinct.
- Please use this page as the cover page for your assignment.

Student 1:

Family Name:

Given Name:

Student #:

Email:

Student 2:

Family Name:

Given Name:

Student #:

Email:

Section to which the assignment will be returned (circle one):

- M (TR11:30-13:00)
- N (W19:00-22:00)
- P (WF16:00-17:30)

Problem	Marking Scheme	Score
1	$16 = 4 \times 4$ marks	
2	$16 = 4 \times 4$ marks	
3	$14 = 4 + 4 + 6$ marks	
4	$18 = 4 + 4 + 4 + 6$ marks	
5	$9 = 3 \times 3$ marks	
6	$27 = 6 + 2 + 3 + 3 + 3 + 5 + 5$ marks	
Total	100 marks	

Assignment 1

Due: Friday, January 23 at 12:00 noon

Please read “How to Prepare Assignment Reports” from the course web page. Make sure your answers are clear and succinct.

1. Let $f : \mathbb{N} \rightarrow \mathbb{R}^+$ and $g : \mathbb{N} \rightarrow \mathbb{R}^+$ be two functions. For each of the following statements determine whether it is true or false. If it is true, give a proof, otherwise give a counterexample.

- (a) If $f(n) \in \Theta(g(n))$, then $2^{f(n)} \in \Theta(2^{g(n)})$,
- (b) if $f(n) \in \Theta(g(n))$, then $\log_2 f(n) \in \Theta(\log_2 g(n))$,
- (c) if $f(n) \in \Theta(g(n))$, then $\sum_{i=1}^n f(i) \in \Theta(\sum_{i=1}^n g(i))$,
- (d) if $(f(n) + g(n))^2 \in \Theta((2g(n))^2)$, then $f(n) \in \Theta(g(n))$.

2. Give a Θ -approximation of the following sums:

- (a) $\sum_{i=1}^n i \cdot 2^i$
- (b) $\sum_{k=5}^{4n} \frac{1}{3k+7}$
- (c) $\sum_{i=2}^n \sum_{j=1}^n \frac{\log i}{j}$
- (d) $\sum_{k=1}^n k^k$

3. Suppose that someone has developed an algorithm to solve a certain problem, which runs in time $T(n, m) \in \Theta(f(n, m))$, where n is the size of the input, and m is a parameter we are free to choose (we can choose it depending on n). In each case determine the value of the parameter $m(n)$ to achieve the (asymptotically) best running time. Justify your answer.

- (a) $f(n, m) = \frac{n^3}{m} + m \cdot n$
- (b) $f(n, m) = \log^3 m + \frac{2^n}{m}$
- (c) $f(n, m) = \frac{8^n n^2}{m} + m \cdot 2^n + m^2$

4. Give asymptotic upper and lower bounds for $T(n)$ in each of the following recurrences. Assume that $T(n)$ is constant for sufficiently small n . Make your bounds as tight as possible, and justify your answers.

- (a) $T(n) = 7T(n/3) + n^2$
- (b) $T(n) = 4T(n/2) + n^2\sqrt{n}$
- (c) $T(n) = T(n-2) + 2\log n$
- (d) $T(n) = T(n/2) + T(n/4) + T(n/8) + n$

5. Suppose that, given constants $a > 1$, $b > 1$, $c > 0$, $d \geq 0$, you have devised two recursive algorithms to solve a particular problem, with recurrence equations

$$T_1(n) = aT_1(n/b) + cn^d$$

$$T_2(n) = bT_2(n/a) + cn^d$$

You wish to select the algorithm with the best asymptotic efficiency.

- (a) Under what conditions would you select Algorithm 1?
 - (b) Under what conditions would you select Algorithm 2?
 - (c) Under what conditions would the choice be immaterial?
6. You are asked to write an efficient algorithm for time-stamping an edge (a transition from one value to another) in a binary time signal. Front-end processing passes your module the signal as an n -element bit array containing exactly one edge (e.g., [00111] for $n = 5$). The edge may be of either positive or negative polarity. The temporal sampling rate is 1GHz.
- (a) Design a sublinear algorithm *findedge* that reports the delay of the edge (in nsec). Use pseudocode to describe your algorithm.
 - (b) Write a recurrence equation for your algorithm.
 - (c) Determine tight asymptotic bounds for your algorithm using
 - i. The method of substitution
 - ii. The recursion tree method
 - iii. The master method
 - (d) You are now asked to write an efficient algorithm for estimating the duration of a single pulse in the binary time signal (e.g., [00110] for $n = 5$). Show that, in general, no sublinear algorithm exists for this problem.
 - (e) Suppose that front-end processing guarantees the pulse to be at least n/a nsec in duration, where $a > 1$ is a constant. Design an algorithm that uses your *findedge* routine to solve the problem in sublinear time. What is the asymptotic running time of your algorithm?