CS21001 Discrete Structures
Tutorial 8

---

1. Find big-Θ estimates for the following positive-real-valued increasing functions $f(n)$.

(a) $f(n) = 125f(n/4) + 2n^3$ whenever $n = 4^t$ for $t \geqslant 1$.
(b) $f(n) = 125f(n/5) + 2n^3$ whenever $n = 5^t$ for $t \geqslant 1$.
(c) $f(n) = 125f(n/6) + 2n^3$ whenever $n = 6^t$ for $t \geqslant 1$.

2. Let the running time of a recursive algorithm satisfy the recurrence

$$T(n) = aT(n/b) + cn^d \log^e n$$

for some $e \in \mathbb{N}$. Let $t = \log_b a$. Deduce the running time $T(n)$ in the big-Θ notation for the three cases: (i) $t < d$, (ii) $t > d$, and (iii) $t = d$.

3. Solve for the following divide-and-conquer recurrence: $T(n) = 2T(n/2) + \dfrac{n}{\log n}$.

4. Let t be the number of one-bits in n. Suppose that the running time of a divide-and-conquer algorithm satisfies the recurrence $T(n) = 2T(n/2) + nt$. When n is a power of 2, we have $t = 1$, so $T(n) = 2T(n/2) + n$. Why does this not imply that $T(n) = \Theta(n \log n)$? Find a correct estimate for $T(n)$ in the big-O notation. (**Remark:** There exist algorithms whose running times depend on t. Example: Left-to-right exponentiation.)

5. Let the running time of a recursive algorithm satisfy the recurrence $T(n) = aT(\sqrt{n}) + h(n)$. Deduce the running time $T(n)$ in the big-Θ notation for the cases: (i) $h(n) = n^d$ for some $d \in \mathbb{N}$, and (ii) $h(n) = \log^d n$ for some $d \in \mathbb{N}_0$.

6. [*Karatsuba multiplication*] You want to multiply two polynomials $a(x)$ and $b(x)$ of degree (or degree bound) $n-1$. Each of the input polynomials is stored in an array of n floating-point variables. The product $c(x) = a(x)b(x)$ is of degree (at most) $2n-2$, and can be stored in an array of size $2n-1$.

(a) Use the school-book multiplication method to compute $c(x)$ (use the convolution formula). Deduce the running time of this algorithm.

(b) Let $t = \lceil n/2 \rceil$. Divide the input polynomials as $a(x) = x^t a_{hi}(x) + a_{lo(x)}$ and $b(x) = x^t b_{hi}(x) + b_{lo(x)}$, where each part of a and b is a polynomial of degree $\leqslant t-1$. But then

$$c(x) = a_{hi}(x)b_{hi}(x)x^{2t} + \Big(a_{hi}(x)b_{lo}(x) + a_{lo}(x)b_{hi}(x)\Big)x^t + a_{lo}(x)b_{lo}(x).$$

The obvious recursive algorithm uses this formula to compute $c(x)$ by making four recursive calls on polynomials of degrees $\leqslant t-1$. Deduce the running time of this algorithm.

(c) Reduce the number of recursive calls to three (how?). Deduce the running time of this algorithm.

7. In the quick-sort algorithm, two recursive calls are made on arrays of sizes i and $n-i-1$ for some $i \in \{0, 1, 2, \ldots, n-1\}$ (assuming that there are no duplicates in the input array). Suppose that all these values of i are equally likely. Deduce the expected running time of quick sort under these assumptions.

8. Suppose that an algorithm, upon an input of size n, recursively solves two instances of size $n/2$ and three instances of size $n/4$. Let the "divide + combine" time be $h(n)$. Find the running times of the algorithm if

(a) $h(n) = 1$, **(b)** $h(n) = n$, **(c)** $h(n) = n^2$, **(d)** $h(n) = n^3$.

9. Use the method of recursion trees to derive the running times of the following algorithms.

(a) Majority finding: $T(n) = T(n/2) + cn$, where c is a positive constant.
(b) Stooge sort: $T(n) = 3T(2n/3) + c$, where c is a positive constant.

10. Deduce the running times of divide-and-conquer algorithms in the big-Θ notation if their running times satisfy the following recurrence relations.

(a) $T(n) = T(2n/3) + T(n/3) + 1$. **(b)** $T(n) = T(2n/3) + T(n/3) + n$.
(c) $T(n) = T(2n/3) + T(n/3) + n\log n$. **(d)** $T(n) = T(2n/3) + T(n/3) + n^2$.

11. Deduce the running times of divide-and-conquer algorithms in the big-Θ notation if their running times satisfy the following recurrence relations.

(a) $T(n) = T(n/5) + T(7n/10) + 1$.
(b) $T(n) = T(n/5) + T(7n/10) + n$.
(c) $T(n) = T(n/5) + T(7n/10) + n \log n$.
(d) $T(n) = T(n/5) + T(7n/10) + n^2$.

12. Consider the following variant of stooge sort for sorting an array A of size n.

1. Recursively sort the first $\lceil 3n/4 \rceil$ elements of A.
2. Recursively sort the last $\lceil 3n/4 \rceil$ elements of A.
3. Recursively sort the first $\lceil n/2 \rceil$ elements of A.

(a) Prove that this algorithm correctly sorts A.
(b) Derive the asymptotic running time of this algorithm.

13. Define two binary operations on $\mathbb{Z}$ as $a \oplus b = a + b + 1$ and $a \odot b = a + b + ab$. Prove that $(\mathbb{Z}, \oplus, \odot)$ is a ring. What are the additive and multiplicative identities in this ring? What are the units in this ring?

14. The set of *Gaussian integers* is defined as $\mathbb{Z}[\mathrm{i}] = \{a + \mathrm{i}b \mid a, b \in \mathbb{Z}\}$. Prove that $\mathbb{Z}[\mathrm{i}]$ is an integral domain. What are the units in this ring? Also define the set $\mathbb{Q}[\mathrm{i}] = \{a + \mathrm{i}b \mid a, b \in \mathbb{Q}\}$. Prove that $\mathbb{Q}[\mathrm{i}]$ is a field.

15. Prove that $\mathbb{Z}[\sqrt{5}] = \{a + b\sqrt{5} \mid a, b \in \mathbb{Z}\}$ is an integral domain. Argue that $\mathbb{Z}[\sqrt{5}]$ contains infinitely many units. Prove that $\mathbb{Q}[\sqrt{5}] = \{a + b\sqrt{5} \mid a, b \in \mathbb{Q}\}$ is a field.

16. Prove that $\mathbb{Z}\left[\frac{1+\sqrt{5}}{2}\right] = \left\{a + \left(\frac{1+\sqrt{5}}{2}\right) b \mid a, b \in \mathbb{Z}\right\}$ is an integral domain. Argue that $\mathbb{Z}\left[\frac{1+\sqrt{5}}{2}\right]$ contains infinitely many units. Prove that $\mathbb{Q}\left[\frac{1+\sqrt{5}}{2}\right] = \left\{a + \left(\frac{1+\sqrt{5}}{2}\right) b \mid a, b \in \mathbb{Q}\right\}$ is a field. Prove/Disprove the following equalities as sets: **(a)** $\mathbb{Z}[\sqrt{5}] = \mathbb{Z}\left[\frac{1+\sqrt{5}}{2}\right]$, **(b)** $\mathbb{Q}[\sqrt{5}] = \mathbb{Q}\left[\frac{1+\sqrt{5}}{2}\right]$.

17. Prove that $\mathbb{Z}[\sqrt{-5}] = \{a + b\sqrt{-5} \mid a, b \in \mathbb{Z}\}$ is an integral domain. Find all the units in this ring. Prove that $\mathbb{Q}[\sqrt{-5}] = \{a + b\sqrt{-5} \mid a, b \in \mathbb{Q}\}$ is a field.

18. Let $n \geqslant 2$, and let $(R_i, +_i, \times_i)$ be rings for $i = 1, 2, 3, \ldots, n$. Define two operations on the Cartesian product $R = R_1 \times R_2 \times \cdots \times R_n$ as $(a_1, a_2, \ldots, a_n) + (b_1, b_2, \ldots, b_n) = (a_1 +_1 b_1, a_2 +_2 b_2, \ldots, a_n +_n b_n)$ and $(a_1, a_2, \ldots, a_n) \cdot (b_1, b_2, \ldots, b_n) = (a_1 \times_1 b_1, a_2 \times_2 b_2, \ldots, a_n \times_n b_n)$ (component-wise operations).

(a) Prove that $(R, +, \cdot)$ is a ring.
(b) If each R_i is commutative, prove that R is commutative too.
(c) If each R_i is with identity, prove that R is with identity too. What are the units of R in this case?
(d) Prove/Disprove: If each R_i is an integral domain, then R is also an integral domain.
(e) Prove/Disprove: If each R_i is a field, then R is also a field.

19. Let R be the set of all functions $\mathbb{Z} \to \mathbb{Z}$. For $f, g \in R$, define $(f+g)(n) = f(n) + g(n)$ and $(fg)(n) = f(n)g(n)$ for all $n \in \mathbb{Z}$.

(a) Prove that R is a commutative ring with identity under these two operations.
(b) What are the units of R?
(c) Is R an integral domain?

20. Let R be the set of all functions $\mathbb{Z} \to \mathbb{Z}$. For $f, g \in R$, define $(f+g)(n) = f(n) + g(n)$ and $(fg)(n) = f(g(n))$ for all $n \in \mathbb{Z}$. Prove/Disprove: R is a ring under these two operations.

21. Let R be the set of all n-bit words for some $n \in \mathbb{N}$. Which of the following is/are ring(s)?

(a) R under bitwise OR and AND operations.
(b) R under bitwise XOR and AND operations.

22. Let R be an integral domain. A non-zero non-unit $p \in R$ is called *prime* in R if $p | (ab)$ implies $p | a$ or $p | b$ (for all $a, b \in R$). A non-zero non-unit $p \in R$ is called *irreducible* if $p = ab$ implies that either a or b is a unit.

(a) What are the primes of $\mathbb{Z}$? What are the irreducible elements of $\mathbb{Z}$?
(b) Prove that every prime is also irreducible.
(c) Demonstrate by an example that all irreducible elements need not be prime.