

1. Find exact closed-form solutions to the following recurrences in two ways (by unwinding and by homogenization), and verify the result by induction.
 - (a) $f(n) = 4f(n - 1) + 9, f(0) = 1.$
 - (b) $f(n) = \frac{3}{5}f(n - 1) + \frac{4}{5}, f(0) = 0.$
 - (c) $f(n) = 3f(n/2) + n, f(1) = 1,$ where n is a power of 2.
2. Find exact closed-form solutions to the following recurrences.
 - (a) $f(n) = 8f(n - 1) - 15f(n - 2), f(0) = 0, f(1) = 1$
 - (b) $f(n) = f(n - 1) + f(n - 2) + 1, f(0) = 0, f(1) = 1$
 (**Hint:** Try homogenizing with $f(n) = g(n) + c$ for some constant c .)
3. Recall that a *saddle* in a table of numbers is an entry that is largest in its column and smallest in its row. In Lecture 2 we showed that every table can have at most one saddle. Here is an algorithm for finding it (if it exists):

Input: A $n \times n$ table T . Assume n is a power of two and all entries of T are distinct.

Algorithm Saddle(T):

If $n = 1$, output the (unique) entry in T .

Otherwise,

Recursively run **Saddle**(T_i) on each of the four quadrants T_1, T_2, T_3, T_4 of T .

Let s_i be the output of **Saddle**(T_i).

Test if s_i is a saddle of T by comparing it to

all numbers in its row and column *except* those in T_i .

If one of $s_1, s_2, s_3,$ or s_4 passes the test, output it.

- (a) Show a sample run of **Saddle** on the following input T :

12	2	5	10
16	7	13	4
15	8	14	9
6	1	11	3

- (b) Let $C(n)$ be the worst-case number of comparisons **Saddle** performs on an $n \times n$ input. Explain why

$$C(n) \leq 4C(n/2) + 4n. \tag{1}$$

- (c) Apply Theorem 6 from Lecture 7 to calculate the big-Oh asymptotic growth of $C(n)$.

- (d) Obtain an exact formula for $C(n)$ assuming the inequality in (1) is an equality. Argue that your solution is an *upper bound* on the number of comparisons performed by **Saddle**.

4. DNA (Deoxyribonucleic acid) is a molecule that carries the genetic instructions for all known organisms and many viruses. It consists of a chain of bases. In DNA chain, there are four types of bases: **A**, **C**, **G**, **T**. For example, a DNA chain of length 10 can be **ACGTACGTAT**.

- (a) Let $g(n)$ be the number of configurations of a DNA chain of length n in which the pairs **TT** and **TG** never appear. Write a recurrence for $g(n)$. (**Hint:** Is the first base a **T**?)
- (b) Solve the recurrence from part (a).
- (c) Which one of the alternatives $g(n) = o(3^n)$, $g(n) = \Theta(3^n)$, or $3^n = o(g(n))$ is correct?