

Analysis of Algorithms

CS 375, Spring 2020

Homework: Recurrences

Due by 11:59pm Anywhere on Earth, **DATE TBD**

- To submit this HW, please email a PDF named `HW_Recurrences_<yourName>` to `eaaron@colby.edu`.
- **READING:** From your textbook (CLRS), please read all of Chapters 2 and 3, and pages 65–67 and 88–97 from Chapter 4. Although not all of it will show up on CS375 exercises, it is all good material to know.
- As you understand, many plans have changed, and I’m no longer giving some exercises that I originally intended to give to cover this material (e.g., an exercise on the maintenance step for invariants). Nonetheless, some students might want additional exercises to solidify their learning, so if you would like some additional *ungraded* exercises on this material, please let me know!
- *A general note for CS375:* When writing up your homework, please write neatly and **explain your answers clearly**, giving all details needed to make your answers easy to understand. Graders may not award full credit to incomplete or illegible solutions.

Exercises

1. Using the *unwinding* method from class, solve the following recurrence relations and give the Θ class of the solution.

(a) $T(n) = T(n - 1) + 5$ for $n > 1$; $T(1) = 0$.

(b) $T(n) = 3T(n - 1)$ for $n > 1$; $T(1) = 4$.

(c) $T(n) = T(n - 1) + n$ for $n > 0$; $T(0) = 0$.

For each of the above, show your work in doing the “unwinding.” To show a full understanding of the unwinding method, please be sure to show a k ’th step—a step at some representative k ’th step in the unwinding, in terms of k rather than some specific number, as demonstrated in the lecture of March 11 and shown in lecture notes—which shows the pattern upon which your solution is based.

Ungraded / Optional: The following two exercises were on the originally posted HW10. We will go over them in class, but **please do not turn them in for HW**.

- $T(n) = T(n/2) + n$ for $n > 1$; $T(1) = 1$. (Assume n is of the form 2^i for this exercise.)
- $T(n) = T(n/3) + 1$ for $n > 1$; $T(1) = 1$. (Assume n is of the form 3^i for this exercise.)

2. Prof. Sue Persmart in the CS Department at Portland Institute of Technology (motto: “Our CS Department is the PIT’s!”) likes to tell a story about the invention of chess.
 - (a) According to legend, the game of chess was invented long ago in India by a certain sage. When he took the invention to his king, the king liked the game so much that he offered the inventor any reward he wanted. The inventor asked for some grain to be obtained as follows: Just one grain of wheat was to be placed on the first square of the chessboard, then two grains on the second square, four grains on the third square, eight grains on the fourth, etc., until all 64 squares had been filled.
If it took 1 second to count each grain of wheat, how long would it take to count all the grains of wheat due to the sage?
 - (b) Imagine that, instead of doubling the number of grains for each square of the chessboard, the inventor asked for adding two grains. Then (assuming again that it took 1 second to count each grain) how long would it take to count all the grains of wheat due to the sage?
3. Solve the following recurrence **showing your work, using the recursion tree method** and give the Θ class of the solution.

$$T(n) = 3T(n - 1) \text{ for } n > 1; T(1) = 4.$$

Please show your work by including a table of the following form. The table should include one row for each level in the tree; be sure to show a representative k ’th level to illustrate the relevant pattern, similar to the k ’th step in the unwinding method.

level	# nodes	work per node	total work
0	nodes on level 0	work per node on level 0	total work on level 0
1	nodes on level 1	work per node on level 1	total work on level 1
2	nodes on level 2	work per node on level 2	total work on level 2
...	...	...	...
k	nodes on level k	work per node on level k	total work on level k
...	...	...	...
$\langle \text{last level} \rangle$	nodes on $\langle \text{last level} \rangle$	work per node on $\langle \text{last level} \rangle$	total work on $\langle \text{last level} \rangle$

Then, explain how you use information from your table to arrive at a final answer. Recall that you can use high-level explanations for Θ analysis; you don’t need to give witnesses for the values of the relevant constants.

NOTES: You might want to explicitly include the second-to-last level in your table, as well as the last level, because the last level is sometimes so different from the remainder of the table that it does not fit into the pattern. Also, you are also welcome to include a picture of the recursion tree as part of your answer if you’d like, but it is not required for the exercise.

4. An algorithm to compute 2^n would be defined by the following specifications:

```
// Input: n, a non-negative integer (i.e., 0 or greater)
// Output:  $2^n$  (i.e., 2 to the n'th power)
```

- (a) Write a recursive algorithm for computing 2^n for any non-negative integer n **that is based on the formula** $2^n = 2^{n-1} + 2^{n-1}$. (This may not be the most natural way to think of an algorithm to compute 2^n , but please be sure to use it for this exercise!)
- (b) Set up a recurrence for the *number of additions made by the algorithm* on input n and solve it **using the recursion tree method**.
As above, please show your work by including a table of the form illustrated in the previous exercise; once again, the table should include one row for each level in the tree and a representative k 'th level to illustrate the relevant pattern, similar to the k 'th step in the unwinding method. Then, explain how you use information from your table to arrive at a final answer. (You are also welcome to include a picture of the recursion tree as part of your answer, if you'd like.)
- (c) What is the Θ complexity class of this algorithm? If the above recurrence is helpful in this, explain why; if not, set up a different recurrence for the runtime of this algorithm and solve it **using a method of your choice** to get the complexity class.
- (d) Is this a good algorithm for computing 2^n ? Please explain your answer! (A couple of sentences could well be a sufficient explanation for this exercise.)
5. Give Θ bounds for the following recurrences. *Be sure to use the Master Theorem for these exercises.* As always, be sure to give a brief explanation of your answers—here, that will include the values for each relevant variable in the Master Theorem, what case of the Theorem you are applying, and a brief explanation of how you know what case to apply.
- (a) $T(n) = 4T(n/2) + n^2$, $T(1) = 1$.
- (b) $T(n) = 4T(n/2) + n^3$, $T(1) = 1$.
6. Give Θ bounds for the following recurrence, using any of the three methods introduced in class (unwinding, recursion tree, Master method).

$$T(n) = 3T(n/2) + n \lg n, T(1) = 1$$

As always, be sure to give a brief explanation (with appropriate details) of your answer.