

General instructions. First read Lecture Notes 1, 2, and 3, including the exercises with answers, then answer the problems given in the rest of this document. Edit this LaTeX template to add your answers, as described in the file named `README.txt`, then compile it to produce `homework_1.pdf`.

Don't know? Say so. If you don't know the answer to a problem or any part of the problem, we encourage you to simply write “*I don't know*” instead of giving an answer. (If you write “I don't know” you can also add any questions you might have.)

Gentle reminder: homework is for learning. As noted in [Lecture Note 1](#), engaging fully with the homeworks is the best way to learn the material. *The primary role of homeworks is to help you learn the material, and the primary role of feedback on homeworks is to help you improve your ideas.*

Please try to explain your ideas as clearly as you can. The clearer your explanations are, the more useful the feedback on them can be.

Collaboration and citation policy. If you are doing this homework as part of a course, follow the course's collaboration and citation policy. In any case, at the end of each answer, list the people you collaborated with, and cite every source that your answer uses ideas from (other than the lecture notes and the sources listed in the lecture notes). If there are none, write “none”.

Problem 1. (Section 1.4) Give the asymptotic values of the following functions, using the Θ -notation. Give the simplest possible answer (e.g., use $\Theta(n^5)$ instead of $\Theta(2n^5)$). You don't have to show your work, but you can if you want feedback on it (and it may help you get partial credit if your answer is wrong).

- (a) $\frac{1}{2}n^3 + 7n^4 - n^3 + 13n$
- (b) $2n^{-2} + 1/(\log^2 n)$
- (c) $n(7n^3 \log^2 n + 9n \log^5 n) + 17^{\log_2 n}$
- (d) $n^2 2^n + 13n^4 + n^3 \log n$
- (e) $n^5 4^n + n 5^n$

Problem 1 answer

- (a) ★ ⋯ ★
- (b) ★ ⋯ ★
- (c) ★ ⋯ ★
- (d) ★ ⋯ ★
- (e) ★ ⋯ ★

Problem 1 collaboration and citations

I collaborated on this problem with the following people:

- ★ ⋯ ★

My answers use ideas from the following sources (other than the lecture notes and textbooks):

- ★ ⋯ ★

Problem 2. (Sections 1.4 and 2.6.1) Use the definition of big-O from Section 1.4 to prove the following theorem:

Theorem 1. Suppose $f(n), g(n), h(n)$ are non-negative functions where $f(n) = O(h(n))$ and $g(n) = O(h(n))$. Then $f(n) \cdot g(n) = O(h^2(n))$.

Give the proof in long form (broken into small steps, just like the long-form proofs in the lecture notes). Be careful to get your reasoning about the quantifiers (in the definitions of big-O) right.

Problem 2 answer

Proof (long form).

1. Consider any functions f, g , and h as described in the theorem.
2. ★...★
3. ★...★
4. Therefore, by the definition of big-O, $f(n) \cdot g(n) = O(h^2(n))$. □

Problem 2 collaboration and citations

I collaborated on this problem with the following people:

● ★...★

My answers use ideas from the following sources (other than the lecture notes and textbooks):

● ★...★

Problem 3. (*Induction*) You encounter the following mysterious piece of code.

```
def C(n):
1. if n = 0: return (0, 1)
2. let (u, w) = C( $\lfloor n/2 \rfloor$ )
3. if n is even: return (u + u, u + w)
4. if n is odd: return (u + w, w + w)
```

- (a) What are the outputs of $C(1)$, $C(2)$, $C(3)$, and $C(4)$? Do not justify your answers.
- (b) Give a long-form proof, by induction on n , that for all $n \geq 1$, executing $C(n)$ does at most $2 + 2 \cdot \log_2 n$ additions. (Here we refer to the additions in Lines 3 and 4 of the algorithm.)
- (c) Give a closed-form expression (in terms of n) for the value output by $C(n)$, assuming n is a non-negative integer. Give a long-form proof, by induction on n , that $C(n)$ indeed outputs that value.

(For examples of inductive long-form proofs, see the proof in Section 4.7.1 of [Lecture Note 4](#), and Exercise 2.9 in Section 2.6.2, whose proof has the right form but contains an error for you to find.)

Problem 3 answer

(a) $C(1) = \star \cdots \star$

$C(2) = \star \cdots \star$

$C(3) = \star \cdots \star$

$C(4) = \star \cdots \star$

(b)

Lemma 1. For all $n \geq 1$, executing $C(n)$ does at most $2 + 2 \log_2 n$ additions.

Proof (long form).

1. The proof is by induction on $n \geq 1$.
2. For the base case $n = 1$, $\star \cdots \star$
- 3.1. For the inductive step, consider any $n \geq 2$, and assume the lemma holds for smaller values.
- 3.2. $\star \cdots \star$
- 3.3. $\star \cdots \star$
- 3.4. So $C(n)$ does at most $2 + 2 \log_2 n$ additions.
4. By Block 3, for any $n \geq 2$, if the lemma holds for smaller values, then it holds for n .
5. By induction (with Step 2), the lemma holds for all n . □

(continued)

(c) (Problem 3, continued)

Lemma 2. For all $n \geq 0$, the value output by $C(n)$ is $\star \cdots \star$

Proof (long form).

1. The proof is by induction on n .
2. For the base case $n = 0$, $\star \cdots \star$
- 3.1. For the inductive step, consider any $n \geq 1$, and assume the lemma holds for smaller values.
- 3.2. Consider the execution of $C(n)$.
- 3.3. $\star \cdots \star$
- 3.4. $\star \cdots \star$
- 3.5.1. Case 1. First consider the case that n is odd.
- 3.5.2. $\star \cdots \star$
- 3.5.3. $\star \cdots \star$
- 3.6.1. Case 2. Otherwise n is even.
- 3.6.2. $\star \cdots \star$
- 3.6.3. $\star \cdots \star$
- 3.7. In either case, $C(n)$ returns $\star \cdots \star$. That is, the lemma holds for this n .
4. By Block 3, for any $n \geq 1$, if the lemma holds for smaller values, then it holds for n .
5. By induction (with Step 2), the lemma holds for all n . □

Problem 3 collaboration and citations

I collaborated on this problem with the following people:

• $\star \cdots \star$

My answers use ideas from the following sources (other than the lecture notes and textbooks):

• $\star \cdots \star$

Problem 4. (Section 3.1)

- (a) Define a specific instance I of Stable Matching with $n = 3$ that has only one stable matching.
- (b) Describe the stable matching.
- (c) Explain clearly why it is stable.
- (d) Explain clearly why no other matching is stable.

Problem 4 answer

- (a) ★ ⋯ ★
- (b) ★ ⋯ ★
- (c) ★ ⋯ ★
- (d) ★ ⋯ ★

Problem 4 collaboration and citations

I collaborated on this problem with the following people:

- ★ ⋯ ★

My answers use ideas from the following sources (other than the lecture notes and textbooks):

- ★ ⋯ ★

Problem 5. (Sections 3.1 and 3.7.1; Lecture Note 2) Recall the algorithm from Figure 3.2:

1. For each pair (i, j) in $\langle (1, 1), (1, 2), (2, 1), (1, 3), (3, 1), (2, 2), (2, 3), (3, 2), (3, 3) \rangle$, do:
 - 1.1. Consider those hospital/doctor pairs (h, d) such that d is h 's i th choice, h is d 's j th choice, and neither h nor d are yet matched. For each such pair (h, d) , match h to d .
2. Return the matched pairs.

Consider the following proof.

Lemma 3. *For any execution of the algorithm, if it returns a perfect matching M , then M is stable.*

Proof (long form).

1. Consider any execution of the algorithm that returns a perfect matching M .
 - 2.1. Consider an arbitrary hospital/doctor pair (h, d) .
 - 2.2.1. Suppose for contradiction that (h, d) is unstable for M .
 - 2.2.2. Let d' and h' be the respective partners of h and d in M (they exist, as M is perfect).
 - 2.2.3. Because (h, d) is an unstable pair, h ranks d above d' .
 - 2.2.4. So, by inspection of the alg. (Line 1), it considered the pair (h, d) before the pair (h, d') .
 - 2.2.5. So, when the algorithm considered the pair (h, d) , hospital h was not yet matched.
 - 2.2.6. Likewise, since (h, d) is unstable for M , doctor d ranks h above h' .
 - 2.2.7. So, by inspection of the alg. (Line 1), it considered the pair (h, d) before the pair (h', d) .
 - 2.2.8. So, when the algorithm considered the pair (h, d) , doctor d was not yet matched.
 - 2.2.9. By Steps 2.2.5 and 2.2.8, when the alg. considered (h, d) , neither h nor d were matched.
 - 2.2.10. So, by inspection of the algorithm, it must have matched h to d in that iteration.
 - 2.2.11. But once a pair is matched, it stays matched, so (h, d) is in M .
 - 2.2.12. This is a contradiction, as no matched pair is unstable.
 - 2.3. By Block 2.2, (h, d) is not unstable for M .
3. By Block 2, no pair is unstable for M . That is, M is stable. □

- (a) What is the first line in this proof that makes an assertion that is not necessarily true?
- (b) What is the logical error at that point in the reasoning?
- (c) Describe an input instance on which the algorithm fails. Use hospitals X, Y, Z and doctors A, B, C . Describe the instance by specifying the 3×3 table of mutual ranks for each pair, as in the solution to Exercise 3.4.
- (d) Describe the execution of the algorithm on that input, and the resulting matching.
- (e) What is the resulting matching?
- (f) What is a pair (h, d) that is unstable for the matching? Why is it unstable?

(continued)

Here is the algorithm for reference:

1. For each pair (i, j) in $\langle (1, 1), (1, 2), (2, 1), (1, 3), (3, 1), (2, 2), (2, 3), (3, 2), (3, 3) \rangle$, do:
 - 1.1. Consider those hospital/doctor pairs (h, d) such that d is h 's i th choice, h is d 's j th choice, and neither h nor d are yet matched. For each such pair (h, d) , match h to d .
2. Return the matched pairs.

Problem 5 answer

(a) ★⋯★

(b) ★⋯★

(c) Here is the input, specified by each pair's mutual ranking:

	A	B	C
X	$(0, 0)$	$(0, 0)$	$(0, 0)$
Y	$(0, 0)$	$(0, 0)$	$(0, 0)$
Z	$(0, 0)$	$(0, 0)$	$(0, 0)$

(d) Here is what happens during the execution of the algorithm on that input:

★⋯★

(e) Here is the resulting matching:

★⋯★

(f) The pair ★⋯★ is unstable, because ★⋯★

Problem 5 collaboration and citations

I collaborated on this problem with the following people:

● ★⋯★

My answers use ideas from the following sources (other than the lecture notes and textbooks):

● ★⋯★