

CSCI 432/532, Spring 2026

Homework 1

Due Monday, January 26 at 9am

Submission Requirements

- Type or clearly hand-write your solutions into a PDF format so that they are legible and professional. Submit your PDF on Gradescope.
- Do not submit your first draft. Type or clearly re-write your solutions for your final submission. If your submission is not legible, we will ask you to resubmit.
- Use Gradescope to assign problems to the correct page(s) in your solution. If you do not do this correctly, we will ask you to resubmit.

Collaboration and Resources

The *best* resources for completing the homework are (in no particular order):

- Your own notes from lectures and problem sessions, and/or lecture recordings.
- The lecture notes linked from the course website for the lecture day.
- The questions channel on Discord.
- Office hours.
- Discussions with classmates.

You may also access almost any resource in preparing your solution to the homework. The exception is that you may not seek answers to the problems on the homework directly, such as by pasting them into a GenAI tool, searching for solutions on a search engine, looking for them on Chegg or similar websites, or asking another person for the solution.

However, you ***must***

- Write your solutions in your own words—this means that you should never be *copying* anything of substance from any source, and
- credit every resource you use, including GenAI tools, by providing links or full chat transcripts. If you use the provided LaTeX template, you can use the `Sources` environment for this. Ask if you need help!

Remember, if you choose to use GenAI tools, not only must you provide a full transcript of your conversation (or a link to one), you must also use the following prompt (or some variation of it), and provide a full transcript of the conversation.

Prompt:

You are acting as a teaching assistant for an advanced undergraduate/graduate algorithms and computer science theory course.

Your role is not to provide full solutions or final answers.

Instead, you should act like we are having a conversation. Ask me a single question and let me respond. Avoid asking me many questions at once or giving me a lot of information at one time. Guide me using hints, leading questions, partial insights, and sanity checks. Help me debug my own proofs, algorithms, or reductions. Do not give a complete solution or full proofs. Be patient with me. Don't give me details so that I can move on to the next part of a problem. Make sure that I understood before moving on.

The course uses Jeff Erickson's Models of Computation lecture notes, so you should guide me to answer problems using the definitions, notation, and style from those notes.

Assume I am a serious student trying to learn, not to shortcut the assignment. I am attaching the assignment, so you should refuse to provide a complete solution to the problems listed. Of course, you can walk me through the "Solved Problems" listed at the end if I ask.

1. Given a string w and a symbol a , let $\text{delete}(a, w)$ be the string w with all instances of a removed. For example, $\text{delete}(\text{z}, \text{jazzy})$ is jay and $\text{delete}(\text{1}, \text{00101110})$ is 0000 . Rubrics for each part of the problems are given below. Note that there are sample solved problems beginning on the next page; **use these as a guide for how to write your solutions.**

- (a) (2 points) Write a recursive function that computes $\text{delete}(a, w)$.

Definition rubric. 2 points =

- + 1 For all correct base cases
- + 1 For all correct recursive cases.

No credit for the rest of the problem unless this part is correct.

- (b) (4 points) For strings x and y and symbol a , prove that $\text{delete}(a, x \cdot y) = \text{delete}(a, x) \cdot \text{delete}(a, y)$.

Induction rubric. 10 points =

- + 1 for explicitly considering an arbitrary object
- + 2 for an explicit valid induction hypothesis
 - Yes, you need to write it down. Yes, even if it's "obvious". Remember that the goal of the homework is to communicate with people who aren't as clever as you.
- + 2 for explicit exhaustive case analysis
 - No credit here if the case analysis omits an infinite number of objects. (For example: all oddlength palindromes.)
 - -1 if the case analysis omits a finite number of objects. (For example: the empty string.)
 - -1 for making the reader infer the case conditions. Spell them out!
 - - No penalty if the cases overlap (for example: even length at least 2, odd length at least 3, and length at most 5.)
- + 1 for proof of cases that do not invoke the inductive hypothesis ("base cases")
 - No credit here if one or more "base cases" are missing.
- + 2 for correctly applying the stated inductive hypothesis
 - No credit here for applying a different inductive hypothesis, even if that different inductive hypothesis would be valid.
- + 2 for other details in cases that invoke the inductive hypothesis ("inductive cases")
 - No credit here if one or more "inductive cases" are missing

- (c) (4 points) Recall the function $\#(a, w)$ from problem session 1, which returns the number of occurrences of the symbol a in string w . For string $w \in \{\text{0}, \text{1}\}^*$, prove that $|\text{delete}(\text{1}, w)| = \#(\text{0}, w)$.

Induction rubric. Same as above.

Solved Problems

1. For any string $w \in \{0, 1\}^*$, let $\text{swap}(w)$ denote the string obtained from w by swapping the first and second symbols, the third and fourth symbols, and so on. For example:

$$\text{swap}(10110001101) = 01110010011.$$

The swap function can be formally defined as follows:

$$\text{swap}(w) := \begin{cases} \varepsilon & \text{if } w = \varepsilon \\ w & \text{if } w = 0 \text{ or } w = 1 \\ ba \bullet \text{swap}(x) & \text{if } w = abx \text{ for some } a, b \in \{0, 1\} \text{ and } x \in \{0, 1\}^* \end{cases}$$

- (a) Prove that $|\text{swap}(w)| = |w|$ for every string w .

Solution: Let w be an arbitrary string.

Assume $|\text{swap}(x)| = |x|$ for every string x that is shorter than w .

There are three cases to consider (mirroring the definition of swap):

- If $w = \varepsilon$, then

$$\begin{aligned} |\text{swap}(w)| &= |\text{swap}(\varepsilon)| && \text{because } w = \varepsilon \\ &= |\varepsilon| && \text{by definition of } \text{swap} \\ &= |w| && \text{because } w = \varepsilon \end{aligned}$$

- If $w = 0$ or $w = 1$, then

$$|\text{swap}(w)| = |w| \quad \text{by definition of } \text{swap}$$

- Finally, if $w = abx$ for some $a, b \in \{0, 1\}$ and $x \in \{0, 1\}^*$, then

$$\begin{aligned} |\text{swap}(w)| &= |\text{swap}(abx)| && \text{because } w = abx \\ &= |ba \bullet \text{swap}(x)| && \text{by definition of } \text{swap} \\ &= |ba| + |\text{swap}(x)| && \text{because } |y \bullet z| = |y| + |z| \\ &= |ba| + |x| && \text{by the induction hypothesis} \\ &= 2 + |x| && \text{by definition of } |\cdot| \\ &= |ab| + |x| && \text{by definition of } |\cdot| \\ &= |ab \bullet x| && \text{because } |y \bullet z| = |y| + |z| \\ &= |abx| && \text{by definition of } \bullet \\ &= |w| && \text{because } w = abx \end{aligned}$$

In all cases, we conclude that $|\text{swap}(w)| = |w|$. ■

Rubric: 5 points: Standard induction rubric (scaled). This is more detail than necessary for full credit.

(b) Prove that $\text{swap}(\text{swap}(w)) = w$ for every string w .

Solution: Let w be an arbitrary string.

Assume $\text{swap}(\text{swap}(x)) = x$ for every string x that is shorter than w .

There are three cases to consider (mirroring the definition of swap):

- If $w = \epsilon$, then

$$\begin{aligned}\text{swap}(\text{swap}(w)) &= \text{swap}(\text{swap}(\epsilon)) && \text{because } w = \epsilon \\ &= \text{swap}(\epsilon) && \text{by definition of } \text{swap} \\ &= \epsilon && \text{by definition of } \text{swap} \\ &= w && \text{because } w = \epsilon\end{aligned}$$

- If $w = \textcolor{red}{0}$ or $w = \textcolor{red}{1}$, then

$$\begin{aligned}\text{swap}(\text{swap}(w)) &= \text{swap}(w) && \text{by definition of } \text{swap} \\ &= w && \text{by definition of } \text{swap}\end{aligned}$$

- Finally, if $w = abx$ for some $a, b \in \{\textcolor{red}{0}, \textcolor{red}{1}\}$ and $x \in \{\textcolor{red}{0}, \textcolor{red}{1}\}^*$, then

$$\begin{aligned}\text{swap}(\text{swap}(w)) &= \text{swap}(\text{swap}(abx)) && \text{because } w = abx \\ &= \text{swap}(ba \bullet \text{swap}(x)) && \text{by definition of } \text{swap} \\ &= \text{swap}(ba \bullet z) && \text{where } z = \text{swap}(x) \\ &= \text{swap}(baz) && \text{by definition of } \bullet \\ &= ab \bullet \text{swap}(z) && \text{by definition of } \text{swap} \\ &= ab \bullet \text{swap}(\text{swap}(x)) && \text{because } z = \text{swap}(x) \\ &= ab \bullet x && \text{by the induction hypothesis} \\ &= abx && \text{by definition of } \bullet \\ &= w && \text{because } w = abx\end{aligned}$$

In all cases, we conclude that $\text{swap}(\text{swap}(w)) = w$. ■

Rubric: 5 points: Standard induction rubric (scaled). This is more detail than necessary for full credit.

2. The *reversal* w^R of a string w is defined recursively as follows:

$$w^R := \begin{cases} \varepsilon & \text{if } w = \varepsilon \\ x^R \bullet a & \text{if } w = a \cdot x \end{cases}$$

A *palindrome* is any string that is equal to its reversal, like **AMANAPLANACANALPANAMA**, **RACECAR**, **POOP**, **I**, and the empty string.

- (a) Give a recursive definition of a palindrome over the alphabet Σ .

Solution: A string $w \in \Sigma^*$ is a palindrome if and only if either

- $w = \varepsilon$, or
- $w = a$ for some symbol $a \in \Sigma$, or
- $w = axa$ for some symbol $a \in \Sigma$ and some *palindrome* $x \in \Sigma^*$.

■

Rubric: 2 points = + 1 for base cases and + 1 for the recursive case. No credit for the rest of the problem unless this part is correct.

- (b) Prove $w = w^R$ for every palindrome w (according to your recursive definition).

You may assume the following facts about all strings x , y , and z :

- Reversal reversal: $(x^R)^R = x$
- Concatenation reversal: $(x \bullet y)^R = y^R \bullet x^R$
- Right cancellation: If $x \bullet z = y \bullet z$, then $x = y$.

Solution: Let w be an arbitrary palindrome.

Assume that $x = x^R$ for every palindrome x such that $|x| < |w|$.

There are three cases to consider (mirroring the definition of “palindrome”):

- If $w = \varepsilon$, then $w^R = \varepsilon$ by definition, so $w = w^R$.
- If $w = a$ for some symbol $a \in \Sigma$, then $w^R = a$ by definition, so $w = w^R$.
- Finally, if $w = axa$ for some symbol $a \in \Sigma$ and some palindrome $x \in P$, then

$$\begin{aligned} w^R &= (a \bullet x \bullet a)^R && \text{because } w = axa \\ &= (x \bullet a)^R \bullet a && \text{by definition of reversal} \\ &= a^R \bullet x^R \bullet a && \text{by concatenation reversal} \\ &= a \bullet x^R \bullet a && \text{by definition of reversal} \\ &= a \bullet x \bullet a && \text{by the inductive hypothesis} \\ &= w && \text{because } w = axa \end{aligned}$$

In all three cases, we conclude that $w = w^R$. ■

Rubric: 4 points: standard induction rubric (scaled)

- (c) Prove that every string w such that $w = w^R$ is a palindrome (according to your recursive definition).

Again, you may assume the following facts about all strings x , y , and z :

- Reversal reversal: $(x^R)^R = x$
- Concatenation reversal: $(x \cdot y)^R = y^R \cdot x^R$
- Right cancellation: If $x \cdot z = y \cdot z$, then $x = y$.

Solution: Let w be an arbitrary string such that $w = w^R$.

Assume that every string x such that $|x| < |w|$ and $x = x^R$ is a palindrome.

There are three cases to consider (mirroring the definition of “palindrome”):

- If $w = \epsilon$, then w is a palindrome by definition.
- If $w = a$ for some symbol $a \in \Sigma$, then w is a palindrome by definition.
- Otherwise, we have $w = ax$ for some symbol a and some non-empty string x .
The definition of reversal implies that $w^R = (ax)^R = x^R a$.
Because x is non-empty, its reversal x^R is also non-empty.
Thus, $x^R = by$ for some symbol b and some string y .
It follows that $w^R = bya$, and therefore $w = (w^R)^R = (bya)^R = ay^R b$.

«At this point, we need to prove that $a = b$ and that y is a palindrome.»

Our assumption that $w = w^R$ implies that $bya = ay^R b$.

The recursive definition of string equality immediately implies $a = b$.

Because $a = b$, we have $w = ay^R a$ and $w^R = aya$.

The recursive definition of string equality implies $y^R a = ya$.

Right cancellation implies $y^R = y$.

The inductive hypothesis now implies that y is a palindrome.

We conclude that w is a palindrome by definition.

In all three cases, we conclude that w is a palindrome. ■

Rubric: 4 points: standard induction rubric (scaled).