HW 1: Extra problems

Instructors: Har-Peled

CS/ECE 374A: Intro. Algorithms & Models of Computation, Fall 2024 Version: **1.0**

Each homework assignment will include at least one solved problem, similar to the problems assigned in that homework, together with the grading rubric we would apply *if* this problem appeared on a homework or exam. These model solutions illustrate our recommendations for structure, presentation, and level of detail in your homework solutions. Of course, the actual *content* of your solutions won't match the model solutions, because your problems are different!

Solved Problems

1 Recall that the *reversal* w^R of a string w is defined recursively as follows:

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

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

1. Give a recursive definition of a palindrome over the alphabet Σ.
2. Prove $w = w^R$ for every palindrome w (according to your recursive definition).
3. Prove that every string w such that $w = w^R$ is a palindrome (according to your recursive definition).

In parts (b) and (c), you may assume without proof that $(x \cdot y)^R = y^R \bullet x^R$ and $(x^R)^R = x$ for all strings x and y.

Solution:

1. A string $w \in \Sigma^*$ is a palindrome if and only if either
   - $w = \epsilon$, 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 = 1/2 for each base case + 1 for the recursive case. No credit for the rest of the problem unless this is correct.

2. 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 three cases in the definition):
   - If $w = \epsilon$, then $w^R = \epsilon$ by definition, so $w = w^R$.
   - If $w = a$ for some symbol $a \in \Sigma$, then $w^R = a$ by definition, so $w = w^R$.

- Suppose $w = axa$ for some symbol $a \in \Sigma$ and some palindrome $x \in P$. Then

$$\begin{aligned} w^R &= (a \cdot x \bullet a)^R && \\ &= (x \bullet a)^R \bullet a && \text{by definition of reversal} \\ &= a^R \bullet x^R \bullet a && \text{You said we could assume this.} \\ &= a \bullet x^R \bullet a && \text{by definition of reversal} \\ &= a \bullet x \bullet a && \text{by the inductive hypothesis} \\ &= w && \text{by assumption} \end{aligned}$$

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

Rubric: 4 points: standard induction rubric (scaled)

3. 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 $y^R a = ya$.
     It immediately follows that $(y^R a)^R = (ya)^R$.
     Known properties of reversal imply $(y^R a)^R = a(y^R)^R = ay$ and $(ya)^R = ay^R$.
     It follows that $ay^R = ay$, and therefore $y = y^R$.
     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).

- No penalty for jumping from $aya = ay^R a$ directly to $y = y^R$.

Rubric:[induction] For problems worth 10 points:

+ 1 for explicitly considering an *arbitrary* object
+ 2 for a valid **strong** induction hypothesis
  - **Deadly Sin!** Automatic zero for stating a weak induction hypothesis, unless the rest of the proof is *perfect*.

- + 2 for explicit exhaustive case analysis
  - No credit here if the case analysis omits an infinite number of objects. (For example: all odd-length palindromes.)
  - −1 if the case analysis omits an finite number of objects. (For example: the empty string.)
  - −1 for making the reader infer the case conditions. Spell them out!
  - No penalty if cases overlap (for example:
- + 1 for 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.

2 (100 PTS.) Ohh, starry recurrence.

2.A. (20 PTS.) As a reminder (or definition, if have never encountered this function before), for $n \geq 0$, we define

$$\log^* n := \begin{cases} 1 & n \leq 1 \\ 1 + \log^*(\log n) & n > 1 \end{cases},$$

where log is in base 2. Prove that for all integers $x > 1$, we have $\log^*(2^x) = 1 + \log^* x$.

For concreteness, compute the exact value of $\log^* N$, where $N = 2^{2^{2^{2^2}}} = 2^{2^{16}} > 10^{6000}$.

Solution:

By definition, we have

$$\log^* 2^x = 1 + \log^* \log(2^x) = 1 + \log^* x.$$

As for the second part, we have

$$\log^* N = \log^* 2^{2^{2^{2^2}}} = 1 + \log^* 2^{2^{2^2}} = 2 + \log^* 2^{2^2} = 3 + \log^* 2^2 = 4 + \log^* 2 = 5 + \log^* 1 = 6.$$

2.B. (80 PTS.) Consider the recurrence

$$T(n) = \begin{cases} 10 & n \leq 10 \\ \frac{n}{\log n} T(\lfloor \log n \rfloor) + n & > 10, \end{cases} \tag{1}$$

where the log is in base 2.

Prove (by induction or the tree method), that $T(n) = O(n \log^* n)$ for all integer $n \geq 1$.

Solution:

Lemma 1.1. *For $n \geq 1$, we have $T(n) \leq 10n \log^* n$.*

Proof: **Base case:** For $n \leq 10$, the claim obviously holds, as $T(n) = 10 \leq 10n \log^* n$, as for $n \leq 10$, $\log^* n \geq 1$, as one can easily verify.

Inductive hypothesis: Assume that $T(n) \leq 10n \log^* n$, for $n \leq k$, where $k \geq 10$ is some integer.

Inductive step: Let $n = k + 1 \geq 11$. Observe that $\lfloor \log n \rfloor < k$, so we can apply the inductive hypothesis to it. We thus have

$$\begin{aligned}
T(n) &= \frac{n}{\log n} T(\lfloor \log n \rfloor) + n && \text{(Definition)}\\
&\leq \frac{n}{\log n} 10 \lfloor \log n \rfloor \log^* \lfloor \log n \rfloor + n && \text{(Inductive hypothesis)}\\
&\leq \frac{n}{\log n} 10 \log n \log^* \log n + n && \text{(Monotonicity)}\\
&= \frac{n}{\cancel{\log n}} 10 \cancel{\log n} (\log^* n - 1) + n\\
&= 10n(\log^* n - 1) + n\\
&= 10n \log^* n - 9n\\
&\leq 10n \log^* n. && \blacksquare
\end{aligned}$$

Questions without solutions.

3 (100 PTS.) Repeat that.

3.A. Let $x_1, \ldots, x_n$ be a sequence of integer numbers, such that $\alpha \leq x_i \leq \beta$, for all i, where α, β are some integer numbers. Prove that there are at least $\lceil n/(\beta - \alpha + 1) \rceil$ numbers in this sequence that are all equal.

3.B. Let $\mathsf{G} = (\mathsf{V}, \mathsf{E})$ be an undirected graph. Unless we say otherwise, a graph has no loops or parallel edges. Prove, using (A), that if $|\mathsf{V}| \geq 2$ there are two distinct nodes u and v such that degree of u is equal to degree of v. Recall that the degree of a node x is the number of edges incident to x.

3.C. Prove that if all vertices in G are of degree at least one, then there is a (simple) path between two distinct nodes u and v such that degree of u is equal to degree of v.

4 (100 PTS.) Mix this.

The *sort*, w^s, of a string $w \in \{0, 1\}^*$ is obtained from w by sorting its characters. For example, $010101^s = 000111$. The sort function is formally defined as follows:

$$w^s := \begin{cases} \epsilon & \text{if } w = \epsilon \\ 0x^s & \text{if } w = 0x \\ x^s 1 & \text{if } w = 1x \end{cases}$$

The ***merge*** function, evaluated in order from top to bottom, is

$$m(x, y) := \begin{cases} y & \text{if } x = \varepsilon \\ x & \text{if } y = \varepsilon \\ 0m(x', y) & \text{if } x = 0x' \\ 0m(x, y') & \text{if } y = 0y' \\ 1m(x', y) & \text{if } x = 1x' \end{cases}$$

For example, we have $m(10, 10) = 1010$, $m(10, 010) = 01010$, and $m(010, 0001100) = 0000101100$.

For a string $x \in \{0, 1\}^*$, let $\#_0(x)$ and $\#_1(y)$ be the number of 0s and 1s in x, respectively. For example, $\#_0(0101010) = 4$ and $\#_1(0101010) = 3$.

4.A. (**Not for submission.**) Prove by induction that for any string $w \in \{0, 1\}^*$ we have that $w^s \in 0^*1^*$.

4.B. Prove by induction that for any string $w \in \{0, 1\}^*$ we have that $\#_0(w) = \#_0(w^s)$. Conclude that $\#_1(w) = \#_1(w^s)$ and $|w| = \left|w^s\right|$, for any string w.

4.C. Prove by induction that for any two strings $x, y \in \{0, 1\}^*$ we have that

$$\#_0(m(x, y)) = \#_0(x) + \#_0(y).$$

(**Hint:** Do induction on $|x| + |y|$.)

Conclude that $\#_1(m(x, y)) = \#_1(x) + \#_1(y)$. and $\left|m(x, y)\right| = \left|x\right| + \left|y\right|$.

[This part is somewhat tedious if you write carefully all the details out explicitly. Avoid repetition by stating that you are (essentially) repeating an argument that was already seen in the proof.]

4.D. Prove by induction that for any two strings x, y of the form 0^*1^*, we have that $m(x, y)$ is of the form 0^*1^*.

4.E. Prove (using the above) that $(x \bullet y)^s = m(x^s, y^s)$ for all strings $x, y \in \{0, 1\}^*$.

5 (100 PTS.) Walk on the grid.

Let $p_0 = (x_0, y_0)$ be a point on the positive integer grid (i.e., x_0, y_0 are positive integer numbers). A point (x, y) is ***good*** if $x = y$ or $x = 0$ or $y = 0$. For a point $p = (x, y)$ its ***successor*** is defined to be

$$\alpha(p) = \begin{cases} (x, y - x - 1) & y > x \qquad \text{(vertical move)} \\ (x - y - 1, y) & x > y \qquad \text{(horizontal move).} \end{cases}$$

Consider the following sequence $W(p_0) = p_0, p_1, \ldots$ computed for p_0. In the ith stage of computing the sequence, if p_{i-1} is good then the sequence is done as we arrived to a good location. Otherwise, we set $p_i = \alpha(p_{i-1})$.

5.A. Prove, by induction, that starting with any point p on the positive integer grid, the sequence $W(p)$ is finite (i.e., the algorithm performs a finite number of steps before stopping).

5.B. (Harder.) Given such a sequence, every step between two consecutive points is either a vertical or a horizontal move. A ***run*** is a maximal sequence of steps in the walk that are the same (all vertical or all horizontal). Prove that starting with a point $p = (x, y)$ there are at most $O(\log x + \log y)$ runs in the sequence $W(p)$.

6 (100 PTS.) Balanced or not.

Let $\Sigma = \{a, b\}$. Consider a string $s \in \Sigma^*$ of length n. The ***depth*** of a string s is $d(s) = \#_a(s) - \#_b(s)$, where $\#_c(s)$ is the number of times the character c appears in s. The maximum depth of a string s is $d_{\max}(s) = \max_{p \text{ is any prefix of } s} d(p)$.

A string $t \in \{a, b\}^*$ is ***weakly balanced*** if $d(t) = 0$. The string t is ***balanced*** if it is weakly balanced, and for any prefix substring p of t, we have that $\#_a(p) \geq \#_b(p)$.

In the following, you can assume that $\forall x, y \in \Sigma^*$, we have $d(xy) = d(x) + d(y)$.

6.A. (20 PTS.) Let $s = s_1 s_2 \ldots s_n$ be the given string. For any i, let $s_{\leq i}$ be the prefix of s formed by the first i characters of s, where $0 \leq i \leq n$. For any i, let $f(i) = d(s_{\leq i})$. Prove that if there are indices i and j, such that $i < j$ and $f(i) = f(j)$, then $s_{i+1} s_{i+2} \ldots s_j$ is a weakly balanced string.

6.B. (40 PTS.) Prove (but not by induction please) that if a string $s \in \Sigma^*$ is balanced, then either:

(i) $s = \epsilon$,

(ii) $s = xy$ where x and y are non-empty balanced strings, or

(iii) $s = axb$, where x is a balanced string.

6.C. (40 PTS.) Prove that for any string $s \in \{a, b\}^*$ of length n, that is balanced, at least one of the following must happen:

(i) The maximum depth of s is $\geq \sqrt{n}$, or

(ii) s can be broken into m non-empty substrings $s = t_1 | t_2 | \cdots | t_m$, such that $t_2, t_3, \ldots t_{m-1}$ are weakly balanced strings, and $m \geq \sqrt{n} - 1$.

For example, the string $abaababaabaabbbaaaabbbbb$ can be broken into substrings

$$a|ba|ab|ab|aabaabbb|aaaabbbb|b$$

Hint: Let $f(i) = d(s_{\leq i})$. Analyze the sequence $f(0), f(1), \ldots, f(n)$, and what happens if the same value repeats in this sequence many times.

6.D. (Harder + not for submission.) Prove that for any string $s \in \Sigma^*$ of length n, that is balanced, with maximum depth $< \sqrt{n}/2$, it must be that s can be broken into $2m + 1$ substrings as follows $s = t_1 t_2 t_3 \ldots t_{2m+1}$, such that the m substrings $t_2, t_4, t_6, \ldots t_{2m}$ are non-empty and balanced. Here m has to be at least $\sqrt{n}/2 - 2$.

7 (100 PTS.) How the first mega tribe was created.

According to an old African myth, in the beginning there were only $n > 1$ persons in the world, and each person formed their own tribe. There were all living in the same forest. Every once in a while two tribes would meet. These meeting tribes would always fight each other to decide which tribe is better, and after a short war, invariably, the tribe with the fewer people (that always lost) would merge into the bigger tribe (if the two tribes were of equal size, one of the tribes would be the losing side). Every person in the tribe that just lost, had to sacrifice a lamb to the forest god, for reasons that remain mysterious, as the lambs did nothing wrong. In the end, only one tribe remained.

Prove, that during this process, at most $n \log_2 n$ lambs got sacrificed. (You can safely assume that no new people were born during this period.)

8 (100 PTS.) A few recurrences.

8.A. Consider the recurrence

$$T(n) = 2n + T(\lfloor n/4 \rfloor) + T(\lfloor (3/4)n \rfloor),$$

where $T(n) = 1$ if $n < 10$. Prove by induction that $T(n) = O(n \log n)$.

8.B. Consider the recurrence

$$T(n) = \begin{cases} T(\lfloor n/2 \rfloor) + T(\lfloor n/6 \rfloor) + T(\lfloor n/7 \rfloor) + n & n \geq 24 \\ 1 & n < 24. \end{cases}$$

Prove by induction that $T(n) = O(n)$.

(An easier proof follows from using the techniques described in section 3 of these notes on recurrences.)

9 (100 PTS.) Yo? (Fall 22).

Let $\Sigma = \{0, 1\}$. A string $w \in \Sigma^*$ is ***yoyo*** if it does not contain the substring 00. Thus $w_1 =$ 01010101111 is yoyo, but 1010101001111 is not. Let Y be the language of all the yoyo strings.

9.A. (50 PTS.) Let $L \subseteq \Sigma^*$ be the language that is the minimal set with the following properties:

- $\varepsilon, 0, 1 \in L$.
- If $x, y \in L$, then $x1y \in L$.

Prove (by strong induction) that $L \subseteq Y$.

9.B. (50 PTS.) Prove (by strong induction) that $Y \subseteq L$ (thus $L = Y$).

10 (100 PTS.) Flipper in the bits (Fall 22).

Let $\Sigma = \{0, 1\}$, and consider the following pair of functions f, g defined over Σ^*:

$$f(w) = \begin{cases} \varepsilon & w = \varepsilon \\ \overline{c} g(x) & w = cx \end{cases} \qquad \text{and} \qquad g(w) = \begin{cases} \varepsilon & w = \varepsilon \\ c f(x) & w = cx, \end{cases}$$

where $c \in \Sigma$, and $\overline{c} = \begin{cases} 1 & c = 0 \\ 0 & c = 1. \end{cases}$ For example, we have

$$f(01100) = 1g(1100) = 11f(100) = 110g(00) = 1100f(0) = 11001g(\varepsilon) = 11001.$$

Similarly, we have $g(01100) = 0f(1100) = 00g(100) = 001f(00) = 0011g(0) = 00110f(\varepsilon) = 00110$.

10.A. (30 PTS.) Give a self contained recursive definition for f that does not involve g.

10.B. (70 PTS.) Prove the following identity for all strings w and x:

$$f(w \bullet x) = \begin{cases} f(w)f(x) & |w| \text{ is even} \\ f(w)g(x) & |w| \text{ is odd.} \end{cases}$$

You may assume without proof any result proved in class, in lab, or in the lecture notes. Otherwise, your proofs must be formal and self-contained, and they must invoke the formal definitions of concatenation $\bullet$, length $|\cdot|$, and the f and g functions. Do not appeal to intuition! In particular, you can use the following claim without proof:

Claim 1.2. *For any string $w = w_1 w_2 \cdots w_n \in \Sigma^*$, with $w_i \in \Sigma$, for all i, and for any index j, $1 \leq j < n$, we have $(w_1 \cdots w_j) \bullet (w_{j+1} \cdots w_n) = w$.*

Solutions without questions

11 24.