HW 1

Due on Wednesday, September 2, 2020 at 10am

CS/ECE 374: Algorithms & Models of Computation, Fall 2020

• Groups of up to three people can submit joint solutions. Each problem should be submitted by exactly one person, and the beginning of the homework should clearly state the Gradescope names and email addresses of each group member. In addition, whoever submits the homework must tell Gradescope who their other group members are.

• Submit your solutions electronically on the course Gradescope site as PDF files. Submit a separate PDF file for each numbered problem. If you plan to typeset your solutions, please use the\LaTeX solution template on the course web site. If you must submit scanned handwritten solutions, please use a black pen on blank white paper and a high-quality scanner app (or an actual scanner, not just a phone camera).

• Please sign up on Gradescope with your real name and your illinois.edu email address. Failing to use your real name and your UIUC email would lead to problems with handling your grades. You can signup to Piazza using a pseudo-name if you want to.

• You may use any source at your disposal – paper, electronic, or human—but you must cite every source that you use, and you must write everything yourself in your own words. See the academic integrity policies on the course web site for more details.

• This semester there would not be any IDK points.

• Avoid the Three Deadly Sins! Any homework or exam solution that breaks any of the following rules will be given an automatic zero, unless the solution is otherwise perfect. Yes, we really mean it. We are not trying to be scary or petty (Honest!), but we do want to break a few common bad habits that seriously impede mastery of the course material.
  – Always give complete solutions, not just examples.
  – Always declare all your variables, in English. In particular, always describe the specific problem your algorithm is supposed to solve.
  – Short complete answers are better than longer answers. Unnecessarily long answers (which by definition are not perfect) would get zero (i.e., 0) points. Avoid empty expressions like “in fact”, “as anybody, or their uncle, can see if they think about it...”, etc.
  – Always give credit to outside sources! (Yes, we are no good with counting.)

See the course web site for more information.

If you have any questions about these policies, please do not hesitate to ask in class, in office hours, or on Piazza.

Extra problems (one fully solved) are available in the HW 1 extra problems collection on the class webpage: https://courses.engr.illinois.edu/cs374/fa2020. It is recommended that you look on these extra problems before doing the homework, since it would help you with doing the homeworks. These are also good practice problems for the midterms and final.
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 $d_{\text{max}}(s) = \max_p$ any prefix of $s$ $d(p)$.

A string $t \in \{a, b\}^*$ is \textit{weakly balanced} if $d(t) = 0$. The string $t$ is \textit{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 $d(xy) = d(x) + d(y)$.

1.A. (20 pts.) Let $s = s_1s_2 \ldots s_n$ 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.

1.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.

1.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 $abaababaabaabbbbaaaabbbbb$ can be broken into substrings

\[ a|ba|ab|ab|aabaabb|aaaabbb|b \]

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.

1.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 $s = t_1t_2t_3\ldots t_{2m+1}$, 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 $\sqrt{n}/2 - 2$.

2. (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.)
A few recurrences.

3.A. Consider the recurrence

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

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

3.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.)