CECS 528, Learning Outcome Assessment 5, Pink, Fall 2023, Dr. Ebert

NO NOTES, BOOKS, ELECTRONIC DEVICES, OR INTERPERSONAL COMMUNICATION ALLOWED. Submit each solution on a separate sheet of paper.

Problem

LO1. Solve the following problems.

(a) Find the multiplicative inverse of 21 mod 58.

(b) For the Strassen-Solovay primality test, is $a = 3$ an accomplice or witness to the fact that $n = 5$ is not prime? Show all work.

LO2. Solve the following problems.

(a) Use the Master Theorem to determine the growth of $T(n)$ if it satisfies the recurrence $T(n) = 4T(n/8) + n^{\log_4 8}$. Defend your answer.

(b) Use the substitution method to prove that, if $T(n)$ satisfies

$$T(n) = 2T(n/2) + 7n \log n$$

then $T(n) = \mathrm{O}(n \log^2 n)$.

LO3. Solve each of the following problems.

(a) Recall the combine step of the `Minimum Distance Pair (MDP)` algorithm where, for each point P in the δ-strip, there is a $2\delta \times \delta$ rectangle whose bottom side contains P and is bisected by the vertical line that divides the points into left and right subsets. Explain why there can be at most 7 other points (from the problem instance) in this rectangle.

(b) Consider the following algorithm called `multiply` for multiplying two n-bit binary numbers x and y, where we assume n is even. Let x_L and x_R be the leftmost $n/2$ and rightmost $n/2$ bits of x respectively. Define y_L and y_R similarly. Let P_1 be the result of calling `multiply` on inputs x_L and y_L, P_2 be the result of calling `multiply` on inputs x_R and y_R, and P_3 the result of calling `multiply` on inputs $x_L + x_R$ and $y_L + y_R$. Then return the value $P_1 \times 2^n + (P_3 - P_1 - P_2) \times 2^{n/2} + P_2$. For the two binary integers $x = 11010101$ and $y = 01101100$, determine the values of P_1, P_2, and P_3 at the root level of recursion, and verify that $xy = P_1 \times 2^n + (P_3 - P_1 - P_2) \times 2^{n/2} + P_2$. Hint: you may evaluate P_1, P_2, and P_3 non-recursively using base-10.

LO4. Solve each of the following problems.

(a) When using the FFT algorithm to compute $\mathrm{DFT}_8^{-1}(6, -5, -3, 4, -8, 2, -1, 10)$, provide a list of all the subproblem instances that must be computed. Hint: there are 15 of them (including the original problem instance) and $\mathrm{DFT}_4^{-1}(-5, 4, 2, 10)$ is one of them.

(b) Compute $\mathrm{DFT}_4(5, 2, 3, -4)$ using the FFT algorithm. Show the solution to each of the seven subproblem instances and, for each one, clearly represent it using DFT notation and apply the formula for computing it. Show all work.

LO5. Answer the following with regards to a correctness-proof outline for the Task Selection algorithm (TSA).

(a) Let $T = t_1, \ldots, t_m$ be the set of non-overlapping tasks selected by TSA and sorted by finish time, i.e. $f(t_i) < f(t_{i+1})$ for all $i = 1, \ldots, m-1$. Let T_{opt} be an optimal set of tasks and assume that, for some $k \geq 1$, $t_1, \ldots, t_{k-1} \in T_{\text{opt}}$, but $t_k \notin T_{\text{opt}}$. Explain why there must be at least one task $t' \in T_{\text{opt}}$ that overlaps with t_k. Hint: "Because if there was no such task …".

(b) Explain why there is *at most* one task $t' \in T_{\text{opt}}$ that overlaps with t_k. Hint: assume there are two overlapping tasks, t' and t'', and explain why this creates a contradiction.

(c) Thus, we can define a new optimal set of tasks $\hat{T}_{\text{opt}} = T_{\text{opt}} - \{t'\} + \{t_k\}$ that contains $t_1, \ldots, t_k$. Continuing in this manner, we may obtain an optimal set of tasks T_{opt} for which $T \subseteq T_{\text{opt}}$. Moreover, we also have $T_{\text{opt}} \subseteq T$, since there is no way of add another task to T that does not overlap with one of T's tasks. For example, explain why it would not be possible to place a task t in between tasks t_i and t_{i+1} for some $i = 1, \ldots, m-1$. Therefore, we have established that $T = T_{\text{opt}}$ and TSA is correct.