St. Francis Xavier University
Department of Computer Science

CSCI 355: Algorithm Design and Analysis
Assignment 3
Due November 21, 2024 at 11:30am

Assignment Regulations.

- This assignment must be completed individually.
- Please include your full name and email address on your submission.
- You may either handwrite or typeset your submission. If your submission is handwritten, please ensure that the handwriting is neat and legible.

[4 marks] 1. For each of the following recurrence relations, give an asymptotic tight bound in terms of n using the master theorem if possible, or otherwise indicate that the master theorem does not apply. Briefly justify your answers.

(a) $T(n) = 7T(n/3) + n^2$.

(b) $T(n) = 3T(n/3) + n/2$.

(c) $T(n) = 16T(n/4) + n$.

(d) $T(n) = 2T(n/2) + n/\log(n)$.

[5 marks] 2. Suppose you receive as input an unsorted array A of size n. All of the elements of A are unique integers between 0 and n inclusive, except for one missing mystery integer.

Describe a divide-and-conquer algorithm that takes as input the array A and an integer n and determines the missing mystery integer in $O(n)$ time. Give a brief justification of your algorithm's runtime.

Note that you can't simply sort the array to find the missing mystery integer, since this would take $\Omega(n \log(n))$ time!

[10 marks] 3. Using Strassen's algorithm, compute the product of the two matrices

$$A = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 1 & 2 & 3 \\ 4 & 5 & 6 & 7 \end{bmatrix} \quad \text{and} \quad B = \begin{bmatrix} 8 & 9 & 1 & 2 \\ 3 & 4 & 5 & 6 \\ 7 & 8 & 9 & 1 \\ 2 & 3 & 4 & 5 \end{bmatrix}.$$

That is, determine the seven intermediate products P_1 through P_7 as well as the final product $C = AB$. You need only make one recursive call, since 2×2 matrices are small enough to multiply directly.

To get you started, the first intermediate product P_1 is

$$P_1 = A_{11} \times (B_{12} - B_{22}) = \begin{bmatrix} 1 & 2 \\ 5 & 6 \end{bmatrix} \times \left(\begin{bmatrix} 1 & 2 \\ 5 & 6 \end{bmatrix} - \begin{bmatrix} 9 & 1 \\ 4 & 5 \end{bmatrix} \right) = \begin{bmatrix} -6 & 3 \\ -34 & 11 \end{bmatrix}.$$

(If you like, you may implement Strassen's algorithm in the programming language of your choice—*in fact, this is encouraged!*—but your answer must include a copy of your source code and a verbose listing of each step of your implementation's output.)

[6 marks] 4. To prepare for the coming exam period, you have decided to pull some all-nighters to increase the amount of studying you can accomplish.

There are n days remaining in the term. For each of these n days, you can either stay awake and study, or sleep and not study. You start things off by sleeping on day 0 so that you are well-rested. Suppose you have an array $S[1..n]$, where $S[i] > 0$ is the number of pages of notes you can study on day i assuming you are well-rested (i.e., slept on day $i-1$).

From past experience, you know that the longer you stay awake, the less productive you are. Each additional day you stay awake reduces the productivity of your studying by half; that is, if you are awake on day i and you last slept on day $i-k$, you can only study $S[i]/2^{k-1}$ pages of notes on that day.

(a) Suppose $n = 6$ and let $S = [3, 5, 5, 2, 2, 4]$. How many pages of notes can you study in total if you sleep only on day 2?

(b) Describe a dynamic programming algorithm that takes as input the array $S[1..n]$ and determines the maximum number of pages of notes you can study. Indicate the time complexity of your algorithm in terms of n. (You do not need to formally prove the time complexity.)

Hint. This is a multiway choice problem. How can we model this problem as a Bellman equation?