Homework 1

Assigned September 20, 2019
Due October 2, 2019 via Canvas

CSCE 823 students have to do all problems for full credit. CSCE 423 students need to do only the Core Problems for full credit, and may do the Advanced Problem for bonus points.

For this homework assignment, you are to work in the team that you established in Homework 0. This will be your collaborative team for the rest of the term. You may freely discuss solutions to exercises within your team, and you are to submit a single pdf file from your team. **The internet is not an allowed resource on this homework!**

Clarity of presentation is important. You should give a clear description of all your algorithms, each with a proof of correctness and analysis of time complexity. You must submit your solutions in a single pdf file via Canvas, and are encouraged to prepare your solutions in LaTeX. **Only** pdf will be accepted, and you should submit only one pdf file. When you submit for Question 1, submit a second pdf file to the *Questions* assignment in Canvas.

Core Problems

1. **(bonus points, but mandatory submission)** Present one question that you have on the lectures and/or textbook on either Order Statistics or Dynamic Programming. This question should be thoughtful and nontrivial, and suggest depth of knowledge in the material. Also, present what you consider to be a reasonable (doesn't have to be completely correct) answer to this problem.

   One way you can come up with good questions is by taking time complexity and correctness into consideration. For example, think about why Select runs in $O(n)$. What kinds of modifications to the algorithm worsen the time complexity? Pick one modification and analyze the new time complexity. Another way is to ask genuine questions on a specific point in the lecture slides. Maybe you are able to follow the correctness proof, until some point, but later got confused. In your answer, reason about the proof up to the point you understand, then describe where you got stuck.

   Your question and answer should be submitted to the *Questions* assignment in Canvas in a pdf file separate from the rest of your homework submission.

2. **(30 Points)** Let $X[1..n]$ and $Y[1..n]$ be two arrays, each containing n numbers already in sorted order. Give an $O(\log n)$-time algorithm to find the median of all $2n$ elements in arrays X and Y. (*Hint:* If an oracle told you what the median is, how would you confirm it?)

3. **(15 points)** Find an optimal parenthesization of a matrix-chain product whose sequence of dimensions is $p = \langle 8, 6, 1, 2, 5 \rangle$. Describe how you arrived at this parenthesization in terms of the algorithm given in class.

4. **(35 points)** A cheese manufacturer hired you to help minimize its distribution costs. Specifically, your employer has a *cheese schedule* $\mathbf{c} = \langle c_1, \ldots, c_n \rangle$, where c_i is the number of pounds of cheese to distribute in week i. To ship the cheese, you may use one of two distributors: WeShipIt4U or UseUs2SendUrStuff. WeShipIt4U ships at a per-pound rate of w, so using WeShipIt4U in week i costs $w\, c_i$. UseUs2SendUrStuff ships at a per-week rate of t, so using

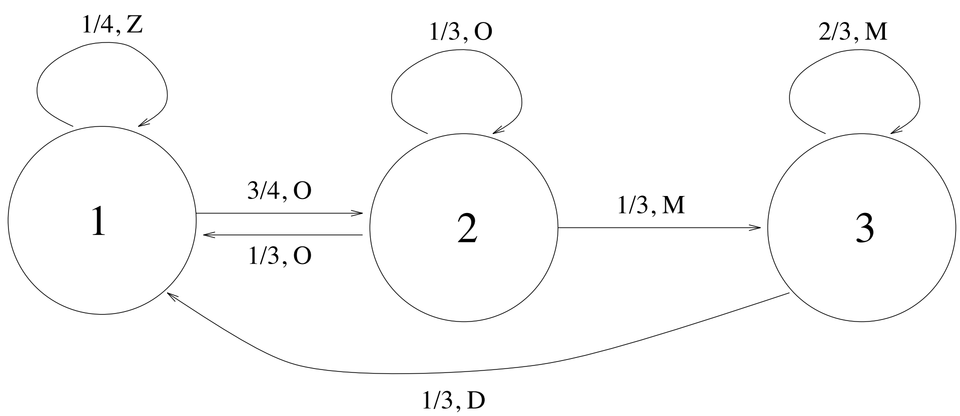
UseUs2SendUrStuff in week i costs t, but to use UseUs2SendUrStuff, you must contract with them for a block of four consecutive weeks.

You may think of a *shipping schedule* $\mathbf{s} = \langle s_1, \ldots, s_n \rangle$, where $s_i = 0$ if you ship with WeShipIt4U in week i and $s_1 = 1$ if you ship with UseUs2SendUrStuff in week i. The cost of a shipping schedule is the total shipping cost for weeks 1 to n, using the cost calculations above. For example, if $\mathbf{c} = \langle 11, 9, 9, 12, 12, 12, 12, 9, 9, 11 \rangle$, $w = 1$, and $t = 10$, then $\mathbf{s} = \langle 0, 0, 0, 1, 1, 1, 1, 0, 0, 0 \rangle$ is optimal with total cost $29 + 40 + 29 = 98$.

Create an efficient dynamic programming algorithm that takes as input $\mathbf{c}$, w, and t and outputs a shipping schedule $\mathbf{s}$ of minimum cost. Argue your algorithm's correctness and time complexity.

Advanced Problem

5. **(35 points)** Consider a k-state probabilistic state machine with states $1, \ldots, k$ and a $k \times k$ transition matrix a_{ij}, where a_{ij} is the probability of the state machine making a transition from state j to state i in one step. (Here $a_{1j} + \cdots + a_{kj} = 1$ for all j.) When the $j \to i$ transition is made, the machine outputs a symbol s_{ij}, which is drawn from some finite alphabet Σ. For example, a simple such probabilistic state machine is shown below:



Consider a situation where the state machine begins in state 1 and makes n transitions emitting the symbols $w_1, \ldots, w_n$.

Describe an efficient dynamic programming algorithm that, given a description of the probabilistic state machine and the string $w_1, \ldots, w_n$, produces a most likely path through the probabilistic state machine that could have produced this sequence of symbols. State the time complexity of your algorithm as a function of both k (the number of states) and n (the length of the output string). Note that the probability of a path through the chain is the product of the probabilities of the transitions taken.