CECS 528, Learning Outcome Assessment 8b, April 7th, Spring 2023, Dr. Ebert

NO NOTES, BOOKS, ELECTRONIC DEVICES, OR INTERPERSONAL COMMUNICATION ALLOWED. Submit solutions to at most 2 LO problems on separate sheets of paper.

Problems

LO4. Use the Master method to determine the growth of $T(n)$ if it satisfies $T(n) = 3T(n/2) + n \log n$.

LO5. Recall that the `Maximum Subsequence Sum (MSS)` problem (Exercise 34) admits a divide-and-conquer algorithm that, on input integer array a, requires computing the maximum of $\text{MSS}_{\text{left}}(a)$, $\text{MSS}_{\text{right}}(a)$, and $\text{MSS}_{\text{mid}}(a)$. If the array is

$$a = -1, 3, -2, 5, -5, 0, 1, 4, 2, -4, -3$$

then

(a) Provide $\text{MSS}_{\text{left}}(a)$, $\text{MSS}_{\text{right}}(a)$. Hint: no need to show work!

(b) Provide the two arrays of sums, Sum_{left} and $\text{Sum}_{\text{right}}$ that are used to compute $\text{MSS}_{\text{mid}}(a)$. Explain and show how to compute $\text{MSS}_{\text{mid}}(a)$ from these two arrays.

LO6. Given that $r = ae + bg$, $s = af + bh$, $t = ce + dg$, and $u = cf + dh$ are the four entries of AB, and Strassen's products are obtained from matrices

$$A_1 = a, B_1 = f - h, A_2 = a + b, B_2 = h, A_3 = c + d, B_3 = e, A_4 = d, B_4 = g - e,$$

$$A_5 = a + d, B_5 = e + h, A_6 = b - d, B_6 = g + h, A_7 = a - c, B_7 = e + f,$$

Compute $P_1, \ldots, P_7$ and use them to compute r, s, t, and u.

LO7. Answer/Solve the following questions/problems.

(a) The dynamic-programming algorithm that solves the `Optimal Binary Search Tree` optimization problem defines a recurrence for the function $\text{wac}(i, j)$. In words, what does $\text{wac}(i, j)$ equal? Hint: do *not* write the recurrence (see Part b).

(b) Provide the dynamic-programming recurrence for $\text{wac}(i, j)$.

(c) Apply the recurrence from Part b to the set of weights 40, 40, 10, 20 (for keys 1-4, respectively). Show the matrix of subproblem solutions and use it to provide an optimal bst.

LO8. Answer/Solve the following questions/problems.

(a) The dynamic-programming algorithm that solves the `Traveling Salesperson` optimization problem (Exercise 30 from the Dynamic Programming Lecture) defines a recurrence for the function $\mathrm{mc}(i, A)$. In words, what does $\mathrm{mc}(i, A)$ equal? Hint: do *not* write the recurrence (see Part b).

Solution.

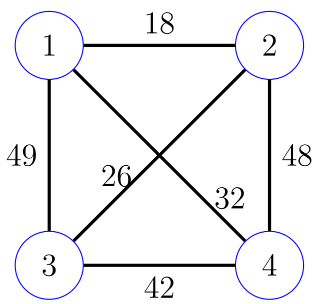
See Solution to Exercise 30 of DP lecture.

(b) Provide the dynamic-programming recurrence for $\mathrm{mc}(i, A)$.

Solution.

See Solution to Exercise 30 of DP lecture.

(c) Apply the recurrence from Part b to the graph below. Show all the necessary computations.



Solution.

We have the following.

$$\mathrm{mc}(1, \{2, 3, 4\}) = \min(18 + \mathrm{mc}(2, \{3, 4\}), 49 + \mathrm{mc}(3, \{2, 4\}), 32 + \mathrm{mc}(4, \{2, 3\}).$$

$$\mathrm{mc}(2, \{3, 4\}) = \min(26 + \mathrm{mc}(3, \{4\}), 48 + \mathrm{mc}(4, \{3\}) = \min(26 + 42, 48 + 48) = 68.$$

$$\mathrm{mc}(3, \{2, 4\}) = \min(26 + \mathrm{mc}(2, \{4\}), 42 + \mathrm{mc}(4, \{2\}) = \min(26 + 48, 42 + 48) = 74.$$

$$\mathrm{mc}(4, \{2, 3\}) = \min(48 + \mathrm{mc}(2, \{3\}), 42 + \mathrm{mc}(3, \{2\}) = \min(48 + 26, 42 + 26) = 68.$$

Therefore,

$$\mathrm{mc}(1, \{2, 3, 4\}) = \min(18 + 68), 49 + 74, 32 + 68) = 86.$$