CECS 528, Learning Outcome Assessment 9b, April 14th, 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

LO5. Consider Karatsuba's algorithm for multiplying two n-bit binary numbers x and y. Let x_L and x_R be the leftmost $\lceil n/2 \rceil$ and rightmost $\lfloor n/2 \rfloor$ 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 $P_1 \times 2^{2\lfloor \frac{n}{2} \rfloor} + (P_3 - P_1 - P_2) \times 2^{\lfloor n/2 \rfloor} + P_2$. Demonstrate Karatsuba's algorithm on $x = 93$ and $y = 72$. Limit your demonstration to the root level of recursion. In other words, do not go deeper than level 0.

LO6. Consider the following algorithm called `Karatsuba's algorithm` for multiplying two n-bit binary numbers x and y. Let x_L and x_R be the leftmost $\lceil n/2 \rceil$ and rightmost $\lfloor n/2 \rfloor$ 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 $P_1 \times 2^{2\lfloor \frac{n}{2} \rfloor} + (P_3 - P_1 - P_2) \times 2^{\lfloor n/2 \rfloor} + P_2$. Show that the algorithm always works by proving in general that $xy = P_1 \times 2^n + (P_3 - P_1 - P_2) \times 2^{n/2} + P_2$ for arbitrary x and y. Hint: to avoid floors and ceilings, you may assume that x and y both have even lengths.

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 wac(i, j). In words, what does wac(i, j) equal? Hint: do *not* write the recurrence (see Part b).

(b) Provide the dynamic-programming recurrence for wac(i, j).

(c) Apply the recurrence from Part b to the keys 1-5 having respective weights 40,20,45,20,50. Show the matrix of subproblem solutions and use it to provide an optimal parenthesization.

LO8. Solve the following problems.

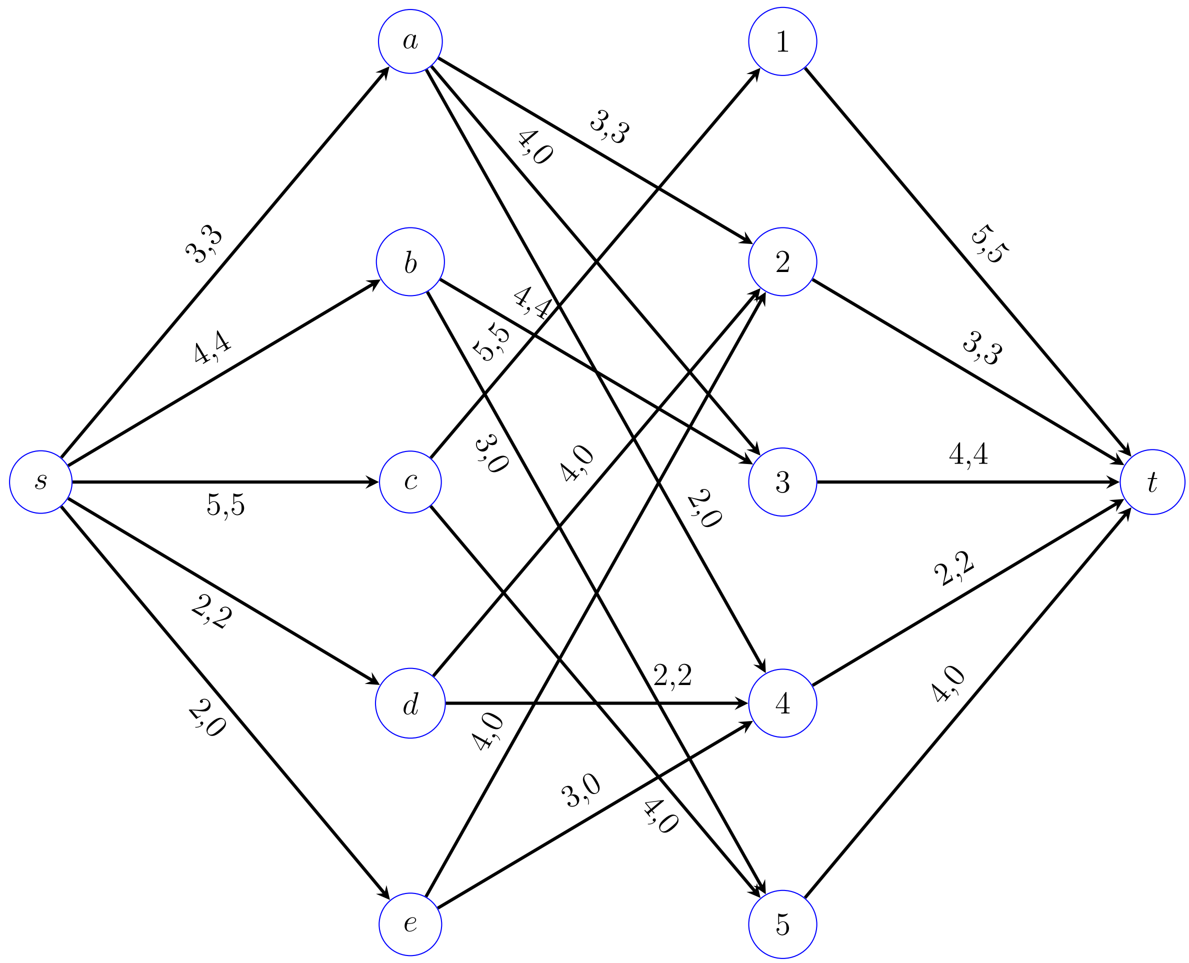
(a) The dynamic-programming algorithm that solves the `Longest Common Subsequence (LCS)` optimization problem defines a recurrence for the function lcs(i, j). In words, what does lcs(i, j) equal? Hint: do *not* write the recurrence (see Part b).

(b) Provide the dynamic-programming recurrence for lcs(i, j).

(c) Apply the recurrence from Part b to the words $u =$ babaab and $v =$ ababba. Show the matrix of subproblem solutions and use it to provide an optimal solution.

LO9. A flow f (2nd number on each edge) has been placed in the network G below.

(a) Draw the residual network G_f and use it to determine an augmenting path P. Highlight path P in the network so that it is clearly visible.



(b) Redraw the original network, but with the f flow values being replaced by the $\Delta(f, P)$ flow values.

(c) What one query is needed to the Reachability-oracle in order to determine if $f_2 = \Delta(f, P)$ is a maximum flow for G?