CECS 528, Learning Outcome Assessment 9a, April 14th, Spring 2023, Dr. Ebert

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 $x = 90$ and $y = 67$. Limit your demonstration to the root level of recursion. In other words, do not go deeper than level 0.

LO6. 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.

(a) Explain why there can be at most 7 other points (from the problem instance) in this rectangle.

(b) Why are those 7 points the only ones for which P's distance must be computed? Defend your answer.

LO7. Answer the following.

(a) The dynamic-programming algorithm that solves the `0-1 Knapsack` optimization problem defines a recurrence for the function $p(i, c)$. In words, what does $p(i, c)$ equal? Hint: do *not* write the recurrence (see Part b). (5 pts)

(b) Provide the dynamic-programming recurrence for $p(i, c)$. (10 pts)

(c) Apply the recurrence from Part b to a knapsack having capacity $M = 11$ and items

item	**weight**	**profit**
1	4	30
2	4	15
3	4	50
4	3	10
5	1	30
6	5	40

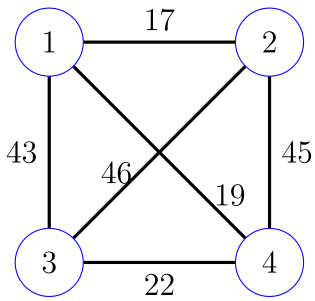
Show the matrix of subproblem solutions and use it to provide an optimal set of items.

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 $\text{mc}(i, A)$. In words, what does $\text{mc}(i, A)$ equal? Hint: do *not* write the recurrence (see Part b).

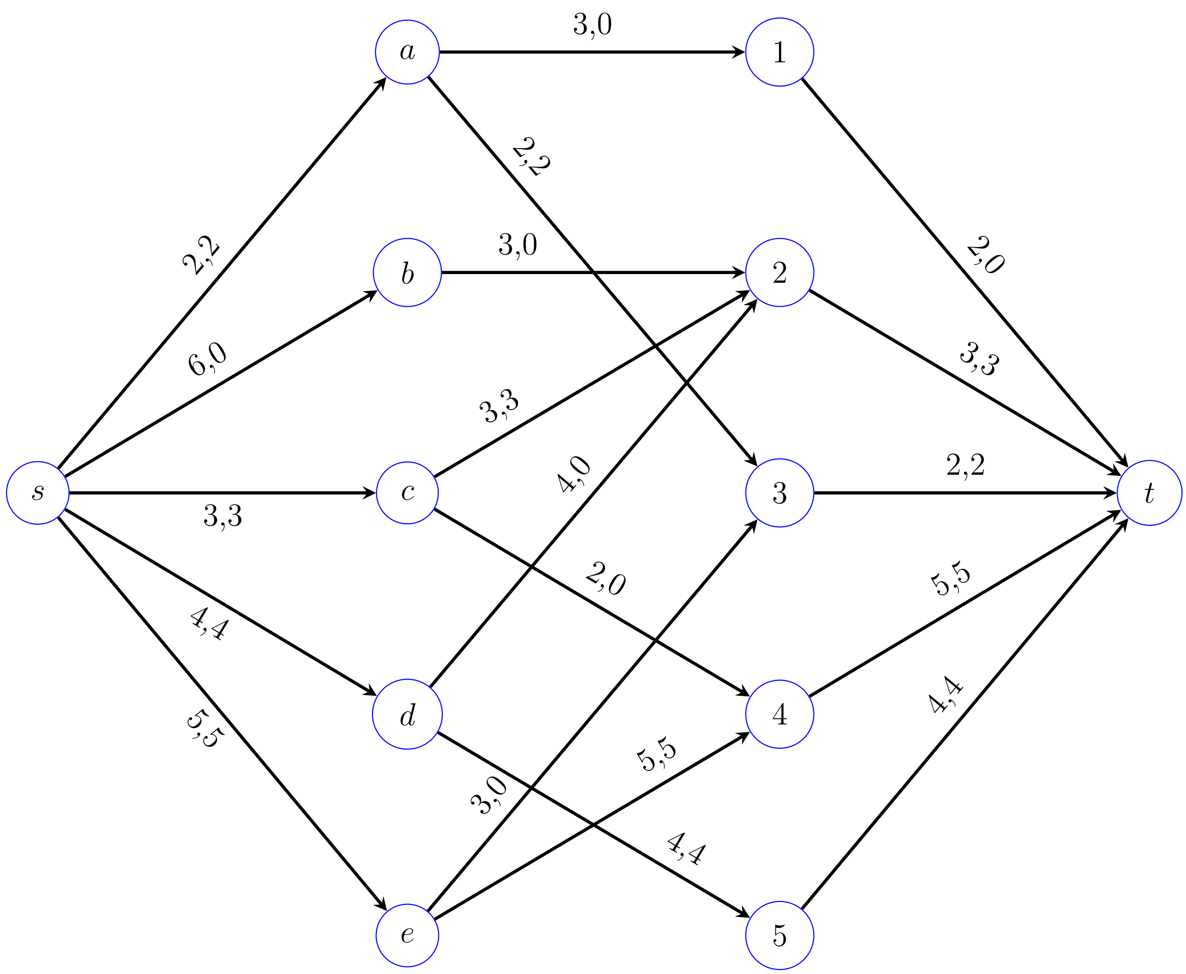
(b) Provide the dynamic-programming recurrence for $\text{mc}(i, A)$.

(c) Apply the recurrence from Part b to the graph below in order to calculate $\text{mc}(1, \{2, 3, 4\})$ Show all the necessary computations.



LO9. A flow f (in red) 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?