Design & Analysis of Algorithms - 344: Homework #4

Due on April 12, 2017 at 11:59pm

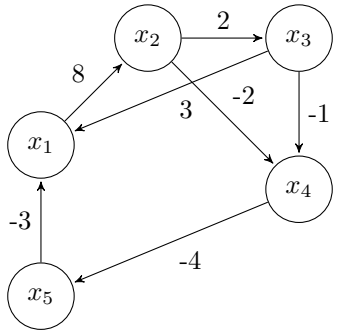
Professor Dr. Kalantari Section #2

Douglas Rudolph, Mustufa Hussain

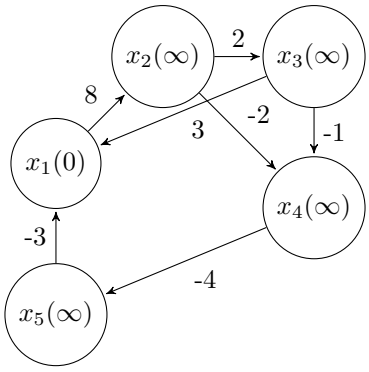
Problem 1 Determine of the following system of inequalities has a feasible solution.

$$\begin{array}{rcl} x_4 - x_2 & \leq & -2 \\ x_1 - x_3 & \leq & 3 \\ x_5 - x_4 & \leq & -4 \\ x_3 - x_2 & \leq & 2 \\ x_2 - x_1 & \leq & 8 \\ x_4 - x_3 & \leq & 1 \\ x_1 - x_5 & \leq & -3 \end{array}$$

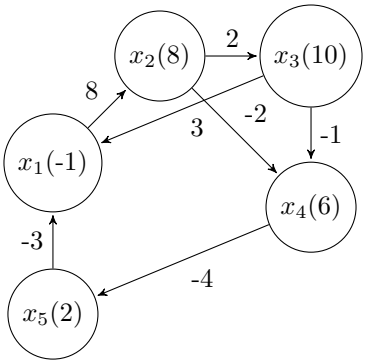
If it has no solution can you change 2 in the first inequality to a number so that the system will be feasible? And what would be the smallest such value to change it to so that the system becomes feasible. Here is the graph that corresponds to the above system:



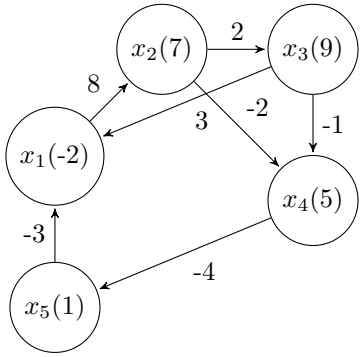
We run the Bellman-Ford on the graph above starting from x_1. Note that () indicates a tentative distance. The graph is initialized as such:



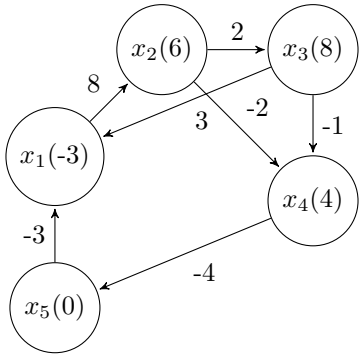
After the first pass:



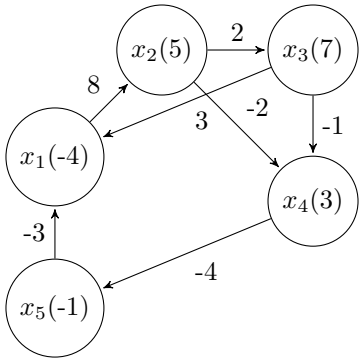
After the second pass:



After the third pass:



After the fourth (and final) pass:



We now check for negative cycles and find that the edge from x_1 to x_2 satisfies the negative cycle condition (that is, 5 ¿ (8 + -4)). As such, we conclude that this graph has a negative cycle, which implies that the system is infeasible.

We see that the negative cycle from x_1 to x_2 to x_4 to x_5 and back to x_1 has a edge sum of (8 + -2 + -4 + -3) = -1. As such, we can change the edge with -2 (the inequality $x_4 - x_2 \leq -2$) to -1 (that is, the inequality becomes $x_4 - x_2 \leq -1$) so that the edge sum becomes non-negative (8 + -1 + -4 + -3 = 0).

Problem 2. Given 4 matrices, $A1, ..., A4$, where Ai is m_{i-1} x m_i matrix, $i = 1, ..., 4$, with $m_0 = 50$, $m_1 = 10$, $m_2 = 30$, $m_3 = 20$, $m_4 = 100$. Compute c(i, j)s by following the matrix chain multiplication algorithm.

When running the matrix chain multiplication algorithm, the goal is to find the best order to multiply some some set of matrices $A_1 * A_2 * ... * A_n$ for a given value n. To accomplish this, we begin by finding the fastest way multiply any 2 matrices that are adjacent to one another and store the result in some type of data structure. After finding the amount of multiplications it would take to multiply any two adjacent matrices, we repeat the same process - but for three adjacent matrices. This process is repeated recursively up until we calculate the most efficient way to multiply a set of n adjacent matrices.

To start let $L = 1$ where L is a variable that refers to the amount adjacent matrices we are multiplying together. Also, let the matrix below be the data structure showing the progression of calculating each matrix for each recursive iteration of the matrix chain algorithm.

In the case that $\mathbf{L = 1}$, the output is as follows. The data in the matrix is showing the amount of multiplications it would take to multiply a given matrix with nothing.

$A_1 * 0$	=	0
$A_2 * 0$	=	0
$A_3 * 0$	=	0

	1	2	3	4
1	0			
2		0		
3			0	
4				0

Now let $\mathbf{L = 2}$. At this step we have to record the amount of multiplications it would take to multiply 2 adjacent matrices. The set of multiplications would be: $M = \{(A_1 * A_2), (A_2 * A_3), (A_3 * A_4)\}$.

$A_1 * A_2$	=	$50 * 10 * 30$	=	$15,000$
$A_2 * A_3$	=	$10 * 30 * 20$	=	$6,000$
$A_3 * A_4$	=	$30 * 20 * 100$	=	$60,000$

	1	2	3	4
1	0	$15,000$		
2		0	$6,000$	
3			0	$60,000$
4				0

Now let $\mathbf{L = 3}$. At this step we have to record the amount of multiplications it would take to multiply 3 adjacent matrices. The set of multiplications would be: $M = \{(A_1)(A_2 * A_3), (A_1 * A_2)(A_3), (A_2 * A_3)(A_4), (A_2)(A_3 * A_4)\}$. (**NOTE**: notice how 1 of the 2 calculations that are being multiplied at this step was already calculated and stored within the matrix from the previous iteration of the matrix chain algorithm.)

$(A_1)(A_2 * A_3)$	=	$50 * 10 * 20 + 6000$	=	$16,000$
$(A_1 * A_2)(A_3)$	=	$15,000 + 50 * 30 * 20$	=	$45,000$
$(A_2 * A_3)(A_4)$	=	$6,000 + 10 * 20 * 100$	=	$26,000$
$(A_2)(A_3 * A_4)$	=	$10 * 30 * 100 + 60,000$	=	$90,000$

	1	2	3	4
1	0	$15,000$	$16,000$	
2		0	$6,000$	$26,000$
3			0	$60,000$
4				0

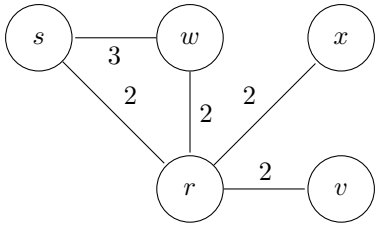
Now let $\mathbf{L = 4}$. At this step we have to record the amount of multiplications it would take to multiply 4 adjacent matrices. The set of multiplications would be: $M = \{(A_1)(A_2 * A_3 * A_4), (A_1 * A_2)(A_3 * A_4), (A_1 * A_2 * A_3)(A_4)\}$.

$(A_1)(A_2 * A_3 * A_4)$	=	$50 * 10 * 100 + 26,000$	=	$76,000$
$(A_1 * A_2)(A_3 * A_4)$	=	$15,000 + 60,000 + 50 * 10 * 100$	=	$175,000$
$(A_1 * A_2 * A_3)(A_4)$	=	$16,000 + 50 * 20 * 100$	=	$116,000$

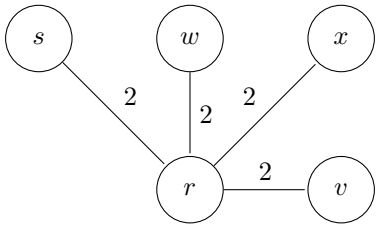
	1	2	3	4
1	0	$15,000$	$16,000$	$76,000$
2		0	$6,000$	$26,000$
3			0	$60,000$
4				0

After running through the algorithm and recursively defining matrix multiplication by calculating *min(c(i-1, j-1)+c(i,j))* at each step, we are able to then calculate all *c(i,j)s*

Problem 3: Construct an undirected graph and place nonnegative weights on its edges so that minimum spanning tree is different from the spanning tree that corresponds to a shortest path tree of one of its vertices. We define the following graph:



The minimum spanning tree of this graph is:



The shortest path tree starting from s is: