NOVA SCHOOL OF BUSINESS & ECONOMICS

2494 - COMPUTATIONAL THINKING & DATA SCIENCE

2019-20, Spring Semester

1st Problem Set

Deadline: February, 24

1. Write a short Python function that takes a positive integer ***n*** and returns the sum of the squares of all the positive integers smaller than ***n***.

2. The Capital Link Inc. is considering 10 investments. The cash required and the streams of cash inflows for each investment are listed in table 1 and 2, respectively. The cash available for investment is $100000. Capital Link wants to find out the investment policy that maximizes its NPV. All cash outflows occur at the beginning of year 1 and all cash inflows occur at the ends of their respective years. The company uses a 10% discount rate for calculating its NPVs.

Investment	1	2	3	4	5	6	7	8	9	10
	$14200	$42700	$14100	$15900	$24200	$44700	$3200	$11000	$26700	$15400

Table 1: Cash required for each investment

Investment / Year	1	2	3	4	5	6	7	8	9	10
1										
2		$13440	$13890	$13330	$9320	$13430				$12610
3		$16130			$10720				$14540	
4		$15320			$10080	$16550			$13230	
5						$14900				
6	$11450	$13820			$8910	$13410				
7	$10530					$12070			$9970	$13910
8	$9690	$12470	$9700			$10860		$16000	$9070	
9				$14630			$9230		$8250	
10		$11260						$14440		

Table 2: Cash flows for each investment

a) Which investments should the company make?

Write a Python program that helps you to solve the problem of Company Link Inc.

Suggestion: Start by implement a function that determines the NPV for each investment.

b) Solve the following modifications of the Capital Link Inc.'s problem by making the appropriate changes in your Python code (solve each part independently of the other):

i) Suppose that if investment 1 is selected, then investment 3 must also be selected.

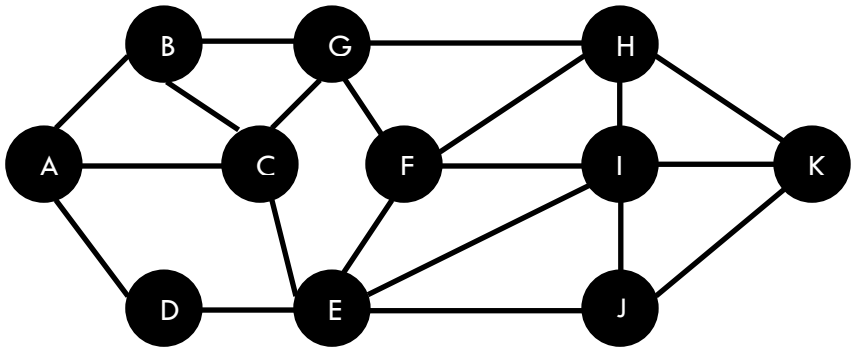
ii) Suppose that at least one of the investments 5 and 6 must be selected.

3. Consider an unweighted graph G = (V, E), where V is the set of nodes and E the set of edges. Suppose that you want to determine the shortest path from node ***s*** to node ***t***, but if it is possible you would like to stop by node ***k*** in the way. However, you only want to pass by node ***k*** if the length of your path doesn't increase more than $\delta\%$.

   Write a Python program that helps you to determine the shortest path between ***s*** and ***t*** given your preference for stopping at node ***k***.

   Note: It should either return the shortest path from ***s*** to ***t*** or the shortest path from ***s*** to ***t*** containing ***k***, depending on the situation.

   Use the following graph to help you testing your program.



   Consider ***s*** = A, ***t*** = K and ***k*** = F.