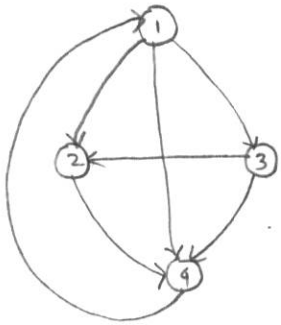
Answers sheet - Final Exam DSE230, June 2016

Name: RAJESH BONDUGULA **PID:** A53108479

Q1 (3pt): (1.) 2. 3. (4.) **Q2 (3pt) :** 1. 2. (3.) 4.
Q3 (10pt): Method 1 is faster. Method 1 has only one reduce and hence can be completed in 1 pass. Method 2 has 2 reduce calls, one for mean calculation another for variance calculation from mean. This does 2 scans on data and hence is slower than Method 1.
Q4 (6pt): s = genderRDD.sample(False, 0.0001) # Sample RDD to get 1008 elements
freq = s.countByValue() # Returns dictionary of frequencies
male_pct = freq['m'] * 100.0 / (freq['m'] + freq['f']) # Male pct
female_pct = freq['f'] * 100.0 / (freq['m'] + freq['f']) # Female pct
Q5 (14pt): grp = graphRDD.groupByKey().mapValues(list) # Grp edges based on starting node.
t = grp.top(1, key=lambda x: len(x[1])) # Get the top node and its connections num
Print "top node =", t[0], "out-degree =", len(t[1])
Q6 (5pt): 1. (2.) 3. 4.
Q7 (6pt): $+0.5 - 0.7 - 0.2 - 0.1 = -0.5$. Sum = -0.5 and hence it is classified as '-1'
Q8 (15pt): % of variance explained by top1 eig vector = $\lambda_1 / (\lambda_1+\lambda_2+\lambda_3)$
% of variance explained by top 2 eig vectors = $(\lambda_1+\lambda_2) / (\lambda_1+\lambda_2+\lambda_3)$
% of variance explained by top 3 eig vectors = $(\lambda_1+\lambda_2+\lambda_3) / (\lambda_1+\lambda_2+\lambda_3) = 1.0$
approximation of x = $\mu + ((x-\mu).v_1)v_1 + ((x-\mu).v_2)v_2$
approximation using v_2 = $x - ((x-\mu).v_3)v_3$
Q9 (8pt): $\frac{\delta}{\delta x}\lvert x-k\rvert = 1$ if $x>k$ else -1 if $x<k$
$\frac{\delta}{\delta x_1}F(\vec{x}) = 1$, $\frac{\delta}{\delta x_2}F(\vec{x}) = -1$, $\frac{\delta}{\delta x_3}F(x) = -1$, $\frac{\delta}{\delta x_4}F(x) = 1$ $\therefore$ gradient = $(1,-1,-1,1)$
Q10 (10pt): 1. (2.) (3.) 4. (5.)

$x = ((x-\mu).v_1)v_1 + ((x-\mu).v_2)v_2 + ((x-\mu).v_3).v_3 + \mu$

Q5



For this example RDD has [(1,2), (1,4), (1,3), (2,4), (3,2), (3,4), (4,1)] (Say this is graphRDD)

= graphRDD.groupByKey().mapValues(list) results [(1,[2,4,3]), (2,[4]), (3,[2,4]), (4,[1])]

zx.top(1, key = lambda x: len(x[1])) will return (1, [2,4,3])

x[0] is top node

len(x[1]) is out degree