CS 3520: Algorithms

Homework 2

**Due Date: Monday, February 18, 2019 at beginning of class**

***Show your steps to receive partial credit.***

.

1. (30 points)
2. Write an algorithm in pseudocode to perform the multiplication of a matrix with a vector .

multiply(A, b, n)

for i ← 0 to n do

for j ← 0 to n do

c [i][0] += A[i][j] \* b[i][0]

return c[i][0]

1. What is the main operation of this algorithm?

Matrix Multiplication

1. How many times is the main operation executed?

=n(n + 1)

= n2 + n

1. What is the efficiency class of this algorithm?

Θ(n2)

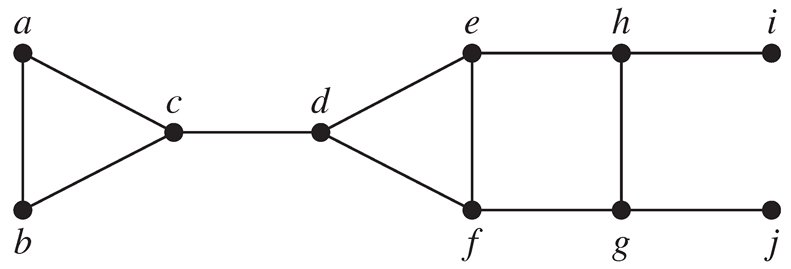
1. (30 points) How many character comparisons will be made by the brute-force algorithm on page 105 of the textbook in searching for the pattern SUCCESS in the following text, which contains 100 characters including letters, spaces and punctuation marks?

START BY DOING WHAT'S NECESSARY; THEN DO WHAT'S POSSIBLE; AND SUDDENLY YOU ARE DOING THE IMPOSSIBLE.

Show your calculation to receive credit. An answer without calculation, even though correct, will not receive credit.

String m - pattern n = 93 minimum comparisons. Plus the comparisons which is 11. 104 comparisons.

1. (40 points) Consider the following graph.



1. Write down the adjacency matrix and adjacency lists specifying this graph. (Assume that the matrix rows and columns and vertices in the adjacency lists follow in the alphabetical order of the vertex labels.)

Adjacency Matrix:   
 a b c d e f g h i j

Adjacency List:

a → b → c

b → a → c

c → a → b → d

d → c → e → f

e → d → f → h

f → d → e → g

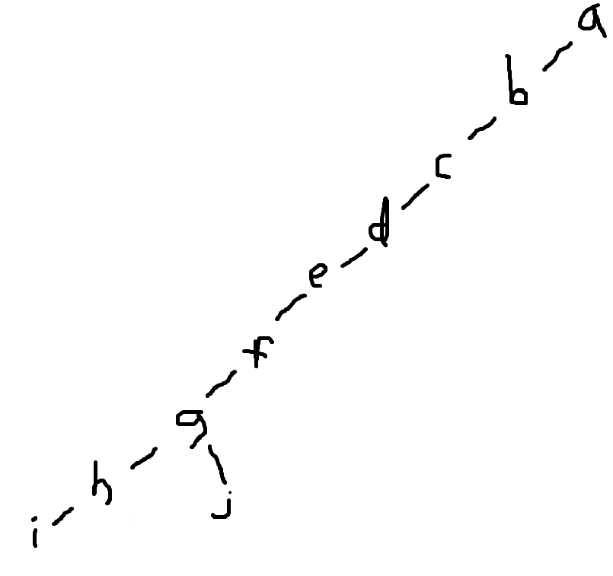
g → f → h → j

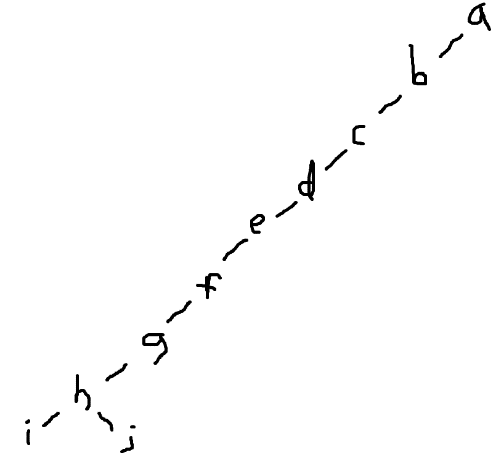
h → e → g → i

i → h

j → g

1. Starting at vertex a and resolving ties by the vertex alphabetical order, traverse the graph by depth-first search and construct the corresponding depth-first search tree. Give the order in which the vertices were reached for the first time (pushed onto the traversal stack) and the order in which the vertices became dead ends (popped off the stack). Draw the resulting forest with solid tree edges and dashed back edges respectively.

Stack: Tree:

 j10, 3

i9, 1

h8, 2

g7, 4

f6, 5

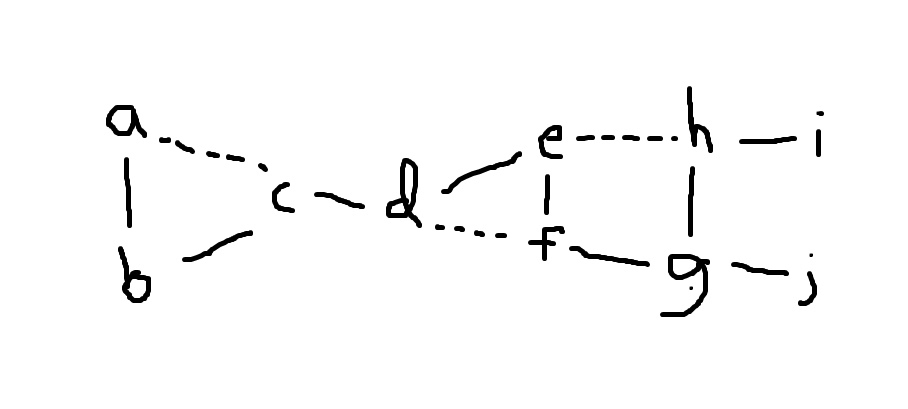
e5, 6

d4, 7

c3, 8

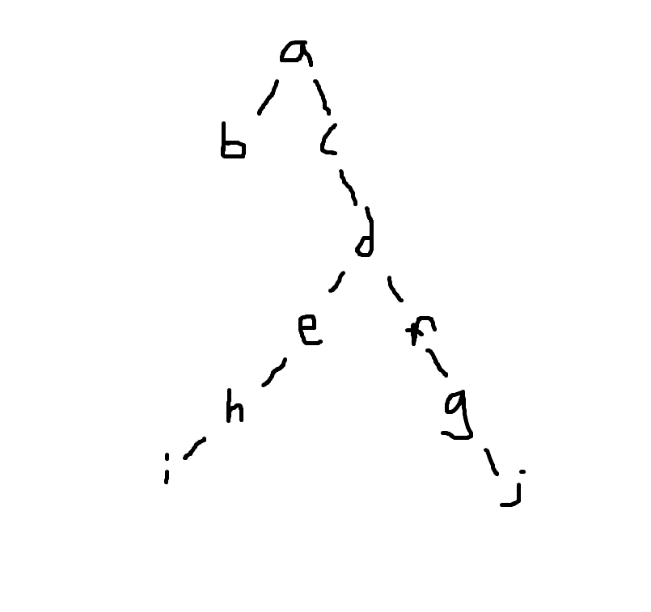
b2, 9

a1, 10

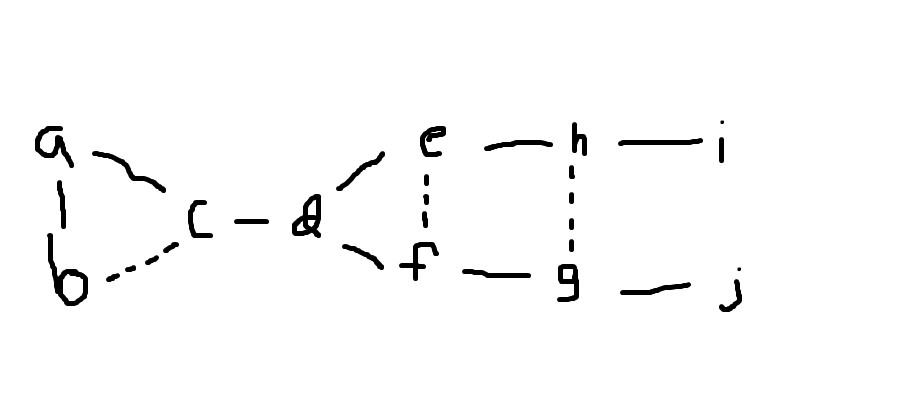


Forest:

1. Traverse the graph by breadth-first search and construct the corresponding breadth-first search tree. Start the traversal at vertex a and resolve ties by the vertex alphabetical order. Draw the resulting forest with solid tree edges and dashed cross edges respectively.



a b c d e f g h i j Tree:



Forest: