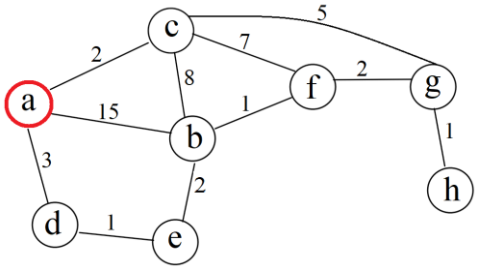
Final:

Due Date: **Monday, May 11, at 6:00pm.**

This exam contains two problems, each asking *multiple* questions. Please answer each question in detail with clear explanation. :)

Problem 1. You are given a random weighted graph. Can you find the shortest distance from **a** as an initial vertex to the other vertices?



A. How would you find the shortest distance? (**Note**: If you have multiple answers in mind, break them apart and explain each one separately.) Explain each solution/algorithm in a few lines.

B. Implement your answer using any programming language you want to.

C. What is the time complexity of your answer? **Explain in detail and show all the work.** (**Note**: If possible, break your code/pseudocode to different parts, calculate the runtime for each step and then try to calculate the total running time based on that.)

Problem 2. Let's say we are given an array of strings and we are interested to see if we can create a cycle using its elements. To create the cycle, string A comes right after string B if the last character of B is same as first character of A.

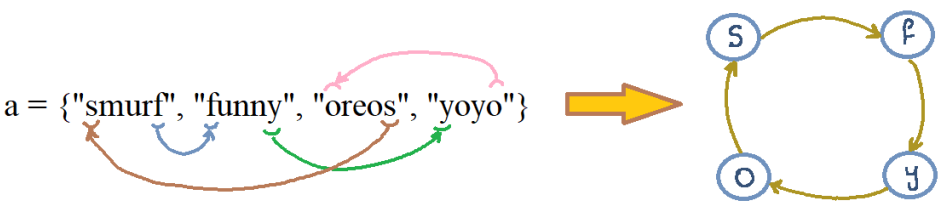
Example 1: *Input*: a = {"smurf", "funny", "oreos", "yoyo"} ➔ *Output*: Yes!

Example 2: *Input*: a = {"soup", "hey", "yes", "push"} ➔ *Output*: Yes!

Example 3: *Input*: a = {"hi", "honey", "yellow", "word"} ➔ *Output*: No!

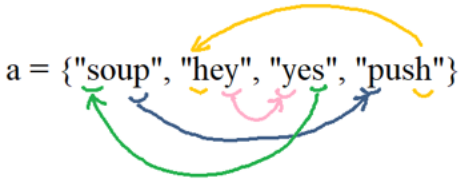
Example 4: *Input*: a = {"giraffe", "cyborg", "eccentric", "oops", "gelato"} ➔ *Output*: No!

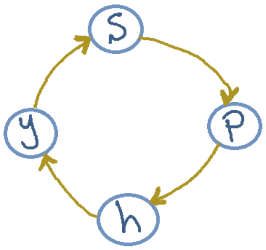
Because:



✓

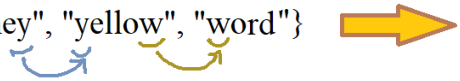
a = {"soup", "hey", "yes", "push"}

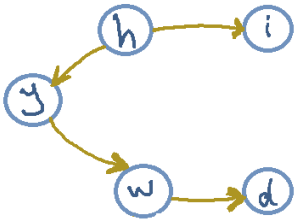




✓

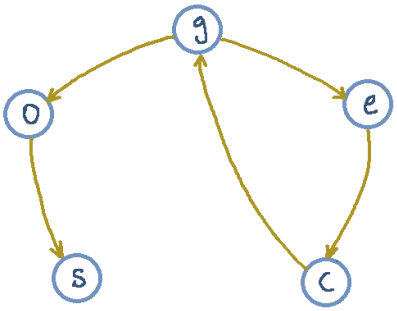
a = {"hi", "honey", "yellow", "word"}





X

a = {"giraffe", "cyborg", "eccentric", "oops", "gelato"}



X

(**Hints**:

i. Each word will create an edge. How?
ii. The in-deg and out-deg of each vertex should be equal. (But this is not the only condition, because a = {“ab” “ba” “xy” “yx”} does not make a cycle. So, hint iii should also be correct)
iii. When finding a back edge in your graph, how could you know if all the vertices are in that cycle? Do you have any pointers that you could help you with that?)

D. How would you decide if the array can form a cycle? (**Note**: If you have multiple answers in mind, break them apart and explain each one separately.) Explain each solution/algorithm in a few lines.

E. Write the pseudocode for the best algorithm you came up with.

F. Implement your answer using any programming language you want to.

G. What is the time complexity of your answer? **Explain in detail and show all the work.** (**Note**: If possible, break your code/pseudocode to different parts, calculate the runtime for each step and then try to calculate the total running time based on that.)

Extra Credit. (2+ points) A number in a sorted array is given to you. Can you find the index of first and last occurrence of the number in the array?

Example 1: *Input*: a = [0, 1, 1, 2, 2, 2, 3, 4, 4], key = 2 ➔ *Output*: start index = 3, end index = 5
Example 2: *Input*: a = [0, 0, 3, 3, 3, 9, 9, 10, 15], key = 10 ➔ *Output*: start index = 7, end index = 7

(**Hints:** Can you find some answer better than linear search? **Make sure the final answer does not become linear**!)

A. How would you find the first and last index of the given number? (**Note**: If you have multiple answers in mind, break them apart and explain each one separately.) Explain each solution/algorithm in a few lines.

B. Write the pseudocode for the best algorithm you came up with.

C. Implement your answer using any programming language you want to.

D. What is the time complexity of your answer? **Explain in detail and show all the work**. (**Note**: If possible, break your code/pseudocode to different parts, calculate the runtime for each step and then try to calculate the total running time based on that.)