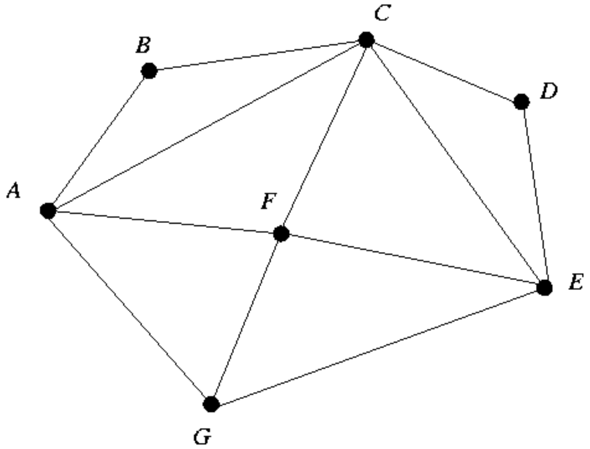
# CS:1210 Practice Problem Set 11

## Complete before Tuesday, 4-21-2015

These practice problems are based on the program that plays the word ladders game. This is called `playLaddersGame2.py` and is posted on the course website.



Consider the network of "words" shown above. Suppose that we call the function `searchWordNetwork` on this word network with source "`A`" and target "`D`".

1. Show the contents of the `reached` dictionary and the `processed` dictionary at the beginning of each iteration of the while-loop in `searchWordNetwork`. Assume that (i) the list of neighbors of each node is in alphabetical order and (ii) each time we pull an element out of `reached` using `popitem()`, *we get the element that is alphabetically largest.*

2. Following up on Problem 1, show the contents of the `processed` dictionary, when it is returned from `searchWordNetwork`.

3. Solve Problem 1 again, but now assume that (i) the list of neighbors of each node is in alphabetical order and (ii) each time we pull an element out of `reached` using `popitem()`, *we get the element that was inserted earliest into `reached`.* The implication of assumption (i) is that the `for`-loops in the function that walk through neighbors will do so in alphabetical order.

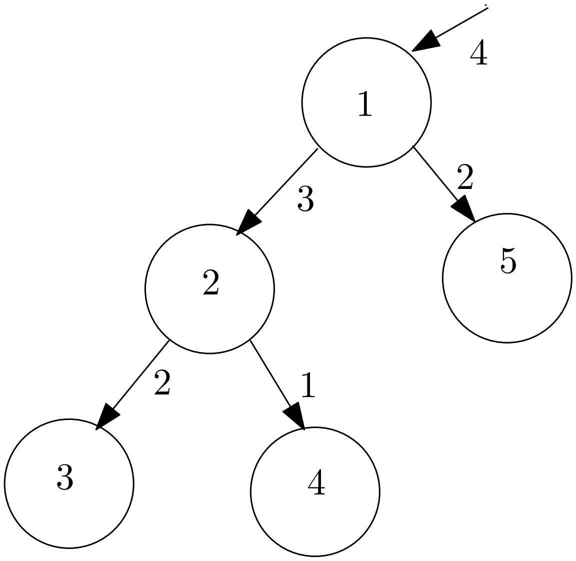
4. Following up on Problem 3, show the contents of the `processed` dictionary, when it is returned from `searchWordNetwork`.

5. This question is about the `fibonacci` function shown below.

```
def fibonacci(n):
    if n == 1 or n == 2:
        return 1

    answer = fibonacci(n-1) + fibonacci(n-2)
    return answer
```

   (a) Here is a picture that shows all the function calls that are made when we call `fibonacci(4)`. Specifically, the picture shows the parameters being sent into each function call (next to each arrow) and also the order in which the functions are called (inside each circle). Draw a larger version of this picture for `fibonacci(6)`. each circle



(b) What output does the function produce, if we insert a `print n` statement as the very first line of the function and call `fibonacci(6)`. You should solve the problem by hand and not by running this function on a computer.

(c) What output does the function produce, if we insert a `print n` statement as the second-last line of the function (just above the `return` statement) and call `fibonacci(6)`. You should solve the problem by hand and not by running this function on a computer.

6. Consider the recursive implementation of the function `power` that we discussed in class (see posted code).

   (a) What output does the function produce, if we insert a `print n` statement as the very first line of the function and execute the function call `power(2, 573)`.

   (b) How many multiplications are performed by the function when we make the function call `power(3, 33)`?