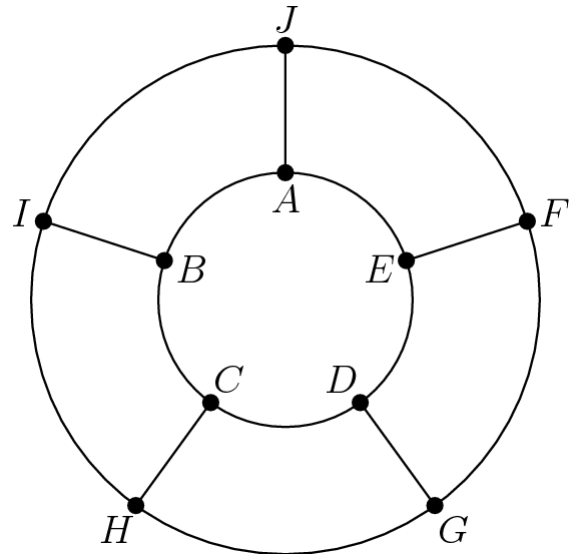


## AI Lab - A Fibonacci relative

Problem 10 at [https://artofproblemsolving.com/wiki/index.php?title=2018\\_AIME\\_I\\_Problems](https://artofproblemsolving.com/wiki/index.php?title=2018_AIME_I_Problems) can be interesting from a search point of view. It essentially asks:

The shown wheel consists of two circles and five spokes, with a label at each point where a spoke meets a circle. A bug walks along the wheel, starting at point *A*. At each step of the process, the bug walks from one labeled point to an adjacent labeled point. Along the inner circle the bug only walks in a counterclockwise direction, and along the outer circle the bug only walks in a clockwise direction. For example, the bug could travel along the path *AJABCHCHIJA*, which has 10 steps. Find the number of paths with 15 steps that begin and end at point *A*.



We will examine this question in the context of BFS. In particular, you will implement the solution to this problem by constructing an appropriate BFS search. The problem will be to find the distribution of how many paths there are to get to point *x* within for each integer less than or equal to some given integer *depth*. Specifically:

1) Do not use the original labeling of the outside wheel. Instead, relabel the wheel so that the outside part of the spokes have the lowercase versions of the letters on the inside. That is, *J* becomes *a*, *I* becomes *b*, *H* becomes *c*, *G* becomes *d*, and *F* becomes *e*.

2) You will run a BFS for *n* levels (where *n* is an arbitrary number and probably not 15) and for each level decide how many ways there are to arrive at each of the 10 positions. For example, at level 0, there is only the bug at position *A*. At level 1, there is 1 way to get to *B* and one way to get to *a*, and 0 ways to get to anywhere else. Since for each position, the bug has a choice of 2 possible ways to go, it means that the sum of all the position possibilities at level *k* is  $2^k$ .

(continued on next page)

3) Your input consists of up to two integers. The first integer will be the number of spokes, defaulting to 5, as in the illustration (ie. the number of distinct letters on the inner wheel). The second input, defaulting to 20, is the number of steps (ie. the level/depth to which to run the BFS). Hint: `chr` may be useful for labelling purposes. `chr(65)` is 'A' and `chr(65+32)` is 'a'. Second hint: First solve the problem for 5 spokes as shown, and then generalize.

4) The output should be a series of  $2 \cdot \text{spokes}$  lines. Each line should start with a letter and indicate the number of ways to reach that letter after exactly  $k$  steps for each value of  $k$  from 0 to the *depth* indicated by the second input. Note that the capital letters come first, in order, and then the lowercase letters. Each column represents one step. For example:

```
D:\Tanitas\TJ\AI\Slider\Code>bugwalk 3 4
Ways to reach 6 spots on a double spoked wheel in 4 steps
A: 1, 0, 1, 1, 3
B: 0, 1, 0, 2, 2
C: 0, 0, 1, 1, 3
a: 0, 1, 0, 2, 2
b: 0, 0, 1, 1, 3
c: 0, 0, 1, 1, 3
Total time: 0.0
```

5) Have fun!