HW #3

Homework exercises should be done individually (You should write the solution by yourself). Solutions must be prepared in python programming language and submitted electronically before **11.59 pm on Sunday, December 18**. No credit will be given to solutions obtained verbatim from the Internet or other sources. **To get full credit for each question, you need to provide a brief explanation of your codes and the efficiency analysis with comments.**

2. Frog A and frog B live in a swamp. Frog A is standing on leaf a_0 and wishes to visit the leaves $a_1, a_2, \dots, a_n$ in this order, by a sequence of direct jumps from leaf to leaf. Similarly, Frog B is standing on leaf b_0 and wishes to visit the leaves $b_1, b_2, \dots, b_m$ in this order. The two frogs have a strong spiritual bond and cannot be more than 1 meter apart. Also, the two frogs refuse to both jump at the same time.

Given the location of each leaf, devise a dynamic programming algorithm that checks if there exists an order of jumps that satisfies the above restrictions. That is, m + n jumps after which frog A is on leaf a_n and frog B is on leaf b_m. The running time should be O(mn).