COMP3121-Ass3-Q3

z5302513, Kihwan Baek

July 2021

1. Subproblem

For every $i \leq n$, we assume that opt(i) is the largest total number of flies that the frog can catch until it gets i lily pad. If a lily pad is not accessible, we set the opt(i) as -1.

2. Base cases and recursion

The base case is opt(1) = f_1 and the frog can go to f_4 or f_5 as the first step. So, $opt(2) = opt(3) = -1$. And also, opt(4) = f_1 + f_4 and opt(5) = f_1 + f_4.

The recursion formula for $i > 5$ is opt(i) = max{opt(i-3), opt(i-4)} + f_i. And also, if the opt(i-3) and opt(i-4) are not accessible from 1 lily pad, the value of opt(i) will be set as -1.

3. How to obtain the final solution

We can get the final solution as getting the last value which can be accessible from 1 lily pad and the order of the last lily pad will be smaller than or equal to 'n'.

4. Time complexity

As following base cases an the recursion case, we need to calculate the largest total number of flies from 1 to n. Therefore, it takes O(n) time.