

COMP3121 21T2 Assignment 3 Q3

Written by Zheng Luo (z5206267)

In order to find the maximum number of files can be caught by the frog, which can only jump forward from i to $i + 3$ or $i + 4$ pad. The position of the frog can be considered as the only variable in this question, hence the solution can be constituted by looping all possible jumping position on the lily pad. Furthermore, the frog has two choices on each lily pad, they can either jump towards $i + 3$ pad, or $i + 4$ pad. Also, assuming $dp(f, pos)$ as the maximum amount of files that can be caught by frog at current position. Hence the maximum amount of files can be determined by comparing the maximum amount of files at current position and the maximum amount of files at next possible lily pad position within a while loop, which is the choice for frog jumping distance. The base case is ending the function if frog has reached the end of lily pad, and return the maximum amount of files at the end of lily pad. The following pseudo code below can explain in a more detailed manner.

```
int calculateMaxFiles(int[n] f) {  
    return dp(int[n] f, 0);  
}  
  
int dp(int[n] f, int pos) {  
    // Base case: End the loop if frog reaches end of lily pad.  
    if (pos >= EndofLilyPad) {  
        return 0;  
    }  
    // Two choices: jump to i+3 or i+4.  
    for (int i = 3; i <= 4; i++) {  
        int maxInNextPosition = dp(f, pos + i);  
        // Select the max between what we got so far,  
        // or future possible choices,  
        // through depth first search.  
        return Math.max(dp(f, pos), (maxInNextPosition + numOffFilesOnCurrentPad));  
    }  
}
```

Overall the time complexity to solve this question is $O(n^2)$, since the algorithm has to consider not only the path from beginning to the end of lily pad, but also the maximum files in different paths.