9101 Assignment 3
Haojin Guo
z5216214

Q3.

3. In a pond there is a sequence on n lily pads arranged in a straight line: $1, 2, 3 \ldots n$. On lily pad i there are $f_i \geq 0$ flies. On lily pad 1 there is a frog sitting. The frog can only jump forward from a lily pad i to either lily pad $i+3$ or lily pad $i+4$. Find the largest number of flies that the frog can catch. (20 pts)
Hint: be careful: not all lily pads are accessible to the frog; the frog can only jump from the starting lily pad 1 to lily pads 4 and 5 but cannot access lily pads 2 and 3. Also, for some i there might be no flies on that lily pad (i.e., $f_i = 0$). So you want to distinguish between lily pads without flies but which are accessible and lily pads which are not accessible.

Solution.
Dynamic programming on the sequence of n lily pads. We will store the maximum number of flies that the flog can catch by jumping to the i_{th} lily pads in a table.

1) Subproblem,
   Assuming that $opt(i)$, where $i \in [1, n]$, represents the largest number of flies that the frog can catch on i_{th} lily pads.
2) Base case,
   $opt(1) = f_1 \quad when\ i = 1, flog\ on\ the\ 1th\ pad,$
   $opt(2) = 0 \quad when\ i = 2, flog\ cannot\ catch\ the\ flies\ on\ 2th\ pad,$
   $opt(3) = 0 \quad when\ i = 3, flog\ cannot\ catch\ the\ flies\ on\ 3th\ pad,$
   $opt(4) = f_1 + f_4 \quad when\ i = 4, flog\ on\ the\ 4th\ pad,$
   $opt(5) = f_1 + f_5 \quad when\ i = 5, flog\ on\ the\ 5th\ pad,$
3) Suppose we have solved all the subproblems for $j < i$ (and meanwhile $i > 4$) and saved them in the table.
   By obtain an optimal solution opt(i) for ith jumping through choosing the maximum value between $opt(i-4) + f_i$ and o$pt(i-3) + f_i$.
   In short, $opt(i) = max\{pt(i-4) + f_i,\ opt(i-3) + f_i\}, and\ i \in [1, n]$.
4) Final result,
   The largest number of files that the flog can catch is maximum value in $opt(i)$ sequence, max $\{opt(i)\}$
5) Time complexity,
   a) Traverse n lily pads to get each value of $opt(i)$. $O(n)$
   b) Traverse the $opt(i)$ sequence, $O(n)$
   Therefore, the total cost is O(n),