 MOISE GABRIEL

QUESTION 1

$f_{k+1} = f_k + f_{k-1}$, $k \geq 1$, $f_0 = 0$, $f_1 = 1$

$F = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}$

(a) We'll prove by induction on k that $F^k = \begin{pmatrix} f_{k+1} & f_k \\ f_k & f_{k-1} \end{pmatrix}$, for all $k \geq 1$.

Base case : $k = 1$

$F^1 = F = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix} = \begin{pmatrix} f_2 & f_1 \\ f_1 & f_0 \end{pmatrix}$, as $f_2 = f_1 + f_0 = 1$

Inductive step

We suppose that $F^k = \begin{pmatrix} f_{k+1} & f_k \\ f_k & f_{k-1} \end{pmatrix}$ and we'll show that $F^{k+1} = \begin{pmatrix} f_{k+2} & f_{k+1} \\ f_{k+1} & f_k \end{pmatrix}$.

$F^{k+1} = F^k \cdot F = \begin{pmatrix} f_{k+1} & f_k \\ f_k & f_{k-1} \end{pmatrix} \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix} = \begin{pmatrix} f_{k+1} + f_k & f_{k+1} \\ f_k + f_{k-1} & f_k \end{pmatrix} = \begin{pmatrix} f_{k+2} & f_{k+1} \\ f_{k+1} & f_k \end{pmatrix}$

Therefore, the induction is complete, so $F^k = \begin{pmatrix} f_{k+1} & f_k \\ f_k & f_{k-1} \end{pmatrix}$ for all $k \geq 1$.

(b) First, the matrix multiplication (a 2×2 matrix is represented here by an array of length 4)

```
def mult (B: Array[int], C: Array [int]) : Array[int] = {
    var R = new Array[int] (4)
    R(0) = B(0) * C(0) + B(1) * C(2)
    R(1) = B(0) * C(1) + B(1) * C(3)
    R(2) = B(2) * C(0) + B(3) * C(2)
    R(3) = B(2) * C(1) + B(3) + C(3)
    return R}
```

And the main where we have $A = (1,0,0,1)$ the identity matrix and $X = F = (1,1,1,0)$ and $n = m$, such that the invariant $F^n = A \cdot X^n$ holds initially:

```
def main (args: Array [String]) = {
    var A = Array (1,0,0,1)
    var X = Array (1,1,1,0)
    var m = scala.io.StdIn.readInt
    var n = m
    // invariant i : F^m = A * X^n
    while (n != 0) {
        if (n % 2 == 0) {X = mult (X,X); n = n/2} // F^n = A * (X^2)^k , where n = 2*k
        else {A = mult (A,X); X = mult (X,X); n = n/2}} // F^n = (A*X) * (X^2)^k, where n = 2*k+1
    // n = 0 & i => F^n = A
    println (A(0)+" "+A(1)+" "+A(2)+" "+A(3)) }
```

As n gets halved after each iteration, the time complexity is $O(\log m)$.

(c) $A = \begin{pmatrix} a+b & a \\ a & b \end{pmatrix} \quad B = \begin{pmatrix} c+d & c \\ c & d \end{pmatrix}$

$$A \cdot B = \begin{pmatrix} ac+bc+ad+bd+ac & ac+bc+ad \\ ac+ad+bc & ac+bd \end{pmatrix} = \begin{pmatrix} (bc+ad+ac)+(ac+bd) & (ac+bc+ad) \\ (ac+bc+ad) & (ac+bd) \end{pmatrix} = \begin{pmatrix} x+y & x \\ x & y \end{pmatrix}$$

for $x = ac+bc+ad$ and $y = ac+bd$

(d)

```
def Fib (m: int) : int = {
    var A = Array (1,0,0,1)
    var X = Array (1,1,1,0)
    var n = m
    while (n! = 0)
    { if (n % 2 == 0) {X = mult (x,x); n = n/2}
      else {A = mult (A,X); X = mult (X,X); n = n/2}}
    return A (2) }
```