260611509
Alexander Chatron-Michaud

Discussed methods w/ Robert Fratila

1) a) base case: $F(1) = 1 \quad F(3\cdot 1) = F(3) = 2 = 2(1)$

$\Rightarrow$ even for $n=1$

assume $F(3n)$ is even

$$F(3(n+1)) = F(3n+3) = F(3n+1) + F(3n+2)$$
$$= F(3n+1) + F(3n) + F(3n+1)$$
$$= 2(F(3n+1)) + F(3n)$$

even + even = even

$\Rightarrow$ since the statement "$F(3n)$ is even" is true for $n=1$ and $n=k+1$, it is true for all n greater than or equal to one

b) base case: $F(0) = \dfrac{\left(\frac{1+\sqrt{5}}{2}\right)^0 - \left(\frac{1-\sqrt{5}}{2}\right)^0}{\sqrt{5}} = 0$ ✓

$F(1) = \dfrac{\left(\frac{1+\sqrt{5}}{2}\right) - \left(\frac{1-\sqrt{5}}{2}\right)}{\sqrt{5}} = \dfrac{\frac{2\sqrt{5}}{2}}{\sqrt{5}} = 1$ ✓

assume $F(n) = \dfrac{\left(\frac{1+\sqrt{5}}{2}\right)^n - \left(\frac{1-\sqrt{5}}{2}\right)^n}{\sqrt{5}}$

let $a = \dfrac{1+\sqrt{5}}{2}$
let $b = \dfrac{1-\sqrt{5}}{2}$

$F(n+1) = F(n) + F(n-1)$ (assumption of $F(n)$)

$$\frac{a^{n+1} - b^{n+1}}{\sqrt{5}} = \frac{a^n - b^n}{\sqrt{5}} + \frac{a^{n-1} - b^{n-1}}{\sqrt{5}}$$

$$\frac{a(a^n) - b(b^n)}{\sqrt{5}} = \frac{a^n(1+a^{-1}) - b^n(1+b^{-1})}{\sqrt{5}}$$

$$\frac{a(a^n) - b(b^n)}{\sqrt{5}} = \frac{a(a^n) - b(b^n)}{\sqrt{5}}$$ ✓

$$1 + a^{-1} = 1 + \frac{2}{1+\sqrt{5}} = \frac{3+\sqrt{5}}{1+\sqrt{5}}\left(\frac{1-\sqrt{5}}{1-\sqrt{5}}\right) = \frac{-2-2\sqrt{5}}{-4} = \frac{1+\sqrt{5}}{2} = a$$

$$1 + b^{-1} = 1 + \frac{2}{1-\sqrt{5}} = \frac{3-\sqrt{5}}{1-\sqrt{5}}\left(\frac{1+\sqrt{5}}{1+\sqrt{5}}\right) = \frac{-2+2\sqrt{5}}{-4} = \frac{1-\sqrt{5}}{2} = b$$

$\Rightarrow$ since the statement $F(0)$ and $F(1)$ prove true and the statement is true for $n+1$, it is true for all $n \geq 0$

2a) Hanoi (m, i, j):

if m > 0 then
  Hanoi (m-1, i, 6-i-j)
  print "Move top ring from rod"+i+"to"+j"
  Hanoi (m-1, 6-i-j, j)

2b) $T(m) = 2(T(m-1)) + 1$

m	T(m)
2	3
3	7
4	15
5	31
6	63

2c) $T(m) = 2^m - 1$

base case(s): $T(0) = 2^0 - 1 = 0$ ✓
$T(1) = 2^1 - 1 = 1$ ✓
$T(2) = 2^2 - 1 = 3$ ✓
$T(3) = 2^3 - 1 = 7$ ✓

Assume $T(m) = 2^m - 1$

we want $T(m+1) = 2^{m+1} - 1$

using 2b, $T(m+1) = 2(T(m)) + 1$

via inductive hypothesis $T(m+1) = 2(2^m - 1) + 1$

$= 2 \cdot 2^m - 2 + 1$

$= 2^{m+1} - 1$ ✓

$\Rightarrow$ since the statement is true for $T(0)$ and $T(m+1)$
the statement is true for $m \geq 0$

3a)

Input: The number n with which the factorial is computed

Precondition: $0 \leq n \leq 20$

Output: $n!$ as a long variable

Postcondition: Output is between 1 and the maximum long.

```
long factorial ← 1
for i ← 1 to n (inclusive) do
    factorial ← i • factorial
return factorial
```

Correctness proof

base case:

$n = 0$ factorial = 1 , loop not executed, return 1 ✓

maintenance: $n = 3$ factorial = 1, loop until factorial = $1 \cdot 2 = 2$, $(2! = 2)$

(true midway)

termination: $n = 3$ factorial = 1, loop: $1 \times 2 \times 3 = 6$, return 6 ✓ $(= 3!)$

correct!

Bin(n, k)

4a) Input: n and k used to find $\binom{n}{k}$

Precondition: n and k are positive integers

Output: $\binom{n}{k}$ as an integer

Post condition: $0 \leq \binom{n}{k} \leq$ max integer value

```
int total ← 0
if  n = k  OR  k = 0
return 1
else if k > n
 return 0
else
total ← Bin(n-1, k) + Bin(n-1, k-1)
return total
```

base case: $\binom{0}{0}$ $\begin{matrix} n=0 \\ k=0 \end{matrix}$ returns 1 ✓

maintenance: $\binom{4}{2}$ $\begin{matrix} n=4 \\ k=2 \end{matrix}$ midway: midway we find = Bin(3,2) + Bin(3,1)

inductive step =7 Bin(2,2) + Bin(2,1) + Bin(3,1)

if we try Bin(2,1), it gives 2 via inductive step = $\binom{2}{1}$ ✓

alg. is true midway

termination: Bin(4,2) returns 6 $= \binom{4}{2}$ ✓

correctness proven

4d) i) The explicit version is faster than the recursive version because it takes less time to do a linear chain of arithmetic expressions than to recursively solve many binomial expressions.

ii) For a fixed n the explicit method time does not change (same number of operations).
Recursively, it is not the same.
Times are longest when the value of the coefficient is higher (max when $k = n/2$) and are the lowest when the coefficient is smaller ($k=0$, $k=n$)
The reason why is because when k is in the middle more recursive instances of the method are called (more calculations)