**Question 1. Graph comparison of 4 algorithms**

**Observation**:

Algorithm2 is the worst algorithm of time execution because time complexity is O(n2)

Algorithm3 is the best one, its time complexity is O(n).

**Question 2**. Proof by Induction

Let F(n) denote the nth Fibonacci number. Prove F(n) > (4/3)n for n > 4.

Hints: Use the fact F(n) = F(n-1) + F(n-2). Since you are using two values, you must prove the two base cases: n = 5 and n = 6.

**Answer**:

F(n) = F(n-1) + F(n-2).

F(0)=0

F(1)=1

F(2)=1

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

F(4)= F(3)+F(2) = 2+1 = 3

Step 1. Prove base case(s).

Let n = 5.

Then LHS = F(5)= F(4)+F(3)=3+2=5 > (4/3)5 =4.21=RHS

Let n = 6.

Then LHS = F(6)= F(5)+F(4)=5+3=8 > (4/3)6=7=RHS

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **n** | 0 | 1 | 2 | 3 | 4 | 5 | 6 |
| **F(n)** | 0 | 1 | 1 | 2 | 3 | 5 | 8 |

Hence the base cases are proved.

Step 2. State induction hypothesis.

Assume the result is true for n = k,

F(k)=F(k-1) + F(k-2) > (4/3)k

Step 3. Induction step.

Prove the result for n = k+1.

F(k+1) = F(k)+F(k-1)

//Using induction hypothesis

F(k) > (4/3)k

F(k-1) > (4/3)(k-1)

> (4/3)k +(4/3)(k-1)

= (4/3)k-1 (4/3 + 1)

= (4/3)k-1 (7/3 \* 3/3)

= (4/3)k-1 (21/9)

> (4/3)k-1 (16/9)

= (4/3)k-1 (4/3)2

= (4/3)k+1

F(k+1)> (4/3)k+1