Topic 7: Induction and Inductive Algorithms

Read: Chapter 5.1, 5.2, Rosen

Consider a class of universal quantification:

$\forall n \in S$, $P(n)$, where $S = \{n_0, n_0+1, n_0+2, \ldots\}$, $n_0 \geq 0$, is a set of non-negative and consecutive integers,

Q: How do we establish the validity of the above universal quantification?

Warning: You can never prove the correctness of a general statement $P(n)$ for all $n \in S$ by just showing that $P(n)$ is true for some n's in S.

Examples:

1. Prove that $1+2+3+\ldots+n = \dfrac{n(n+1)}{2}$, for all integers $n \in N$.

   Observe that S is the set of all natural numbers N with $n_0 = 1$.

2. The Generalized De Morgan's Laws:

   Prove that

   $$\neg(p_1 \wedge p_2 \wedge \ldots \wedge p_n) \equiv \neg p_1 \vee \neg p_2 \vee \ldots \vee \neg p_n,$$
   $$\neg(p_1 \vee p_2 \vee \ldots \vee p_n) \equiv \neg p_1 \wedge \neg p_2 \wedge \ldots \wedge \neg p_n,\ n \geq 2.$$

   Observe that $S = N - \{1\}$ with $n_0 = 2$.

3. Prove that

$$P(n): 1*2 + 2*3 + \ldots + n(n+1) = \frac{n(n+1)(n+2)}{3}, \forall n \geq 1.$$

Observe that S is the set of all natural numbers N with $n_0 = 1$.

4. Prove that P(n): $\sum_{i=0}^{n} ar^i = \frac{a(r^{n+1}-1)}{r-1}, r \neq 1, n \in N \cup \{0\}$.

Observe that S is the set of all non-negative integers with $n_0 = 0$.

5. Prove that $2^n > n^2$, $\forall n \in N, n > 4$.

Observe that $S = N - \{1, 2, 3, 4\}$ with $n_0 = 5$.

6. The Fundamental Theorem of Arithmetic:

Prove that $\forall\ n \in N, n > 1$, n is either a prime or can be written as a product of primes.

Observe that $S = N - \{1\}$ with $n_0 = 2$.

In order to establish the validity of a given universal quantification P with domain of disclosure S defined above, we use a powerful proof technique known as (mathematical) induction.

(Mathematical) Induction:

Theorem: Given a universal quantification

P(n): $\forall n \in S$, P(n), where $S = \{n_0, n_0+1, n_0+2, \ldots\}$, $n_0 \geq 0$, is a set of non-negative and consecutive integers.

If $P(n_0)$ is true, and if $\forall k \geq n_0$, $P(k) \rightarrow P(k+1)$ is true, then the universal quantification $\forall\, n \in S$, $P(n)$ must also be true. Hence,

$$(P(n_0) \wedge (\forall\, k \geq n_0, P(k) \rightarrow P(k+1))) \rightarrow (\forall\, n \in S, P(n)).$$

Proof: TBA.

Proving P(n) using Induction:

Two steps:

1. Basis step:

   We must prove that $P(n_0)$ is true.

2. Inductive step:

   We must prove that $\forall k \geq n_0$, if $P(k)$ is true, then $P(k+1)$ must also be true.
   (The assumption that $P(k)$ is true is called the ***inductive hypothesis***.)

Remarks:

1. When using induction, you must specify the domain of disclosure of n clearly.
2. The domain of disclosure S must be a set of non-negative and consecutive integers in the form $\{n_0, n_0+1, n_0+2, \ldots\}$, with $n_0 \geq 0$ being the smallest integer in S. If S is not in this proper form, then induction cannot be applied to P(n) directly.
3. There may be more than one base case in the quantification P(n). If so, one must prove the validity of each and every possible base case in the basis step.
4. You must state the inductive hypothesis clearly and indicate when it is being used in the proof.
5. For any given arbitrary integer $k \geq n_0$, one must prove that if P(k) is true, then P(k+1) must also be true.
6. After proving that $P(k) \rightarrow P(k+1)$, one must state the conclusion established by using induction.

Proving Universal Quantifications using Induction:

1. Prove that P(n): $1 + 2 + \ldots + n = \dfrac{n(n+1)}{2}, \forall n \in N.$

   Proof: We will prove that P(n) is true by using induction on n, n ∈ N.

   Basis step: When $n = 1, \dfrac{n(n+1)}{2} = \dfrac{1*2}{2} = 1.$

   Hence, P(1) is true.

   Inductive step: Assuming that P(k) is true, we have

   $1 + 2 + \ldots + k = \dfrac{k(k+1)}{2}, \forall k \geq 1.$

   We must now prove that P(k+1) is also true. Hence, we need to prove that

   $$1 + 2 + \ldots + k + (k+1) = \frac{(k+1)(k+2)}{2}.$$

   Observe that

   $$\begin{aligned} & 1 + 2 + \ldots + k + (k+1) \\ & = (1 + 2 + \ldots + k) + (k+1) \\ & = \frac{k(k+1)}{2} + (k+1) \qquad \text{(Inductive hypothesis)} \\ & = \frac{(k+1)(k+2)}{2}. \end{aligned}$$

   Hence, if P(k) is true, then P(k+1) must also be true. By induction, P(n) must be true for all n ∈ N.

2. Prove the Generalized De Morgan's Laws:

$\neg(p_1 \wedge p_2 \wedge \ldots \wedge p_n) \equiv \neg p_1 \vee \neg p_2 \vee \ldots \vee \neg p_n,$ (*)

$\neg(p_1 \vee p_2 \vee \ldots \vee p_n) \equiv \neg p_1 \wedge \neg p_2 \wedge \ldots \wedge \neg p_{n,}\ n \geq 2.$

Proof: We will prove P(n) given in (*) by using induction on n, $n \in N - \{1\}$. Hence, $n_0 = 2$.

Basis step: When n = 2, we can verify easily that $\neg(p_1 \wedge p_2) \Leftrightarrow \neg p_1 \vee \neg p_2$ using a truth table. Hence, P(2) is true.

Inductive step: Assuming that P(k) is true, we have $\neg(p_1 \wedge p_2 \wedge \ldots \wedge p_k) \equiv \neg p_1 \vee \neg p_2 \vee \ldots \vee \neg p_k,\ \forall k \geq 2.$ We must now prove that P(k+1) is also true. Hence, we need to prove that

$$\neg(p_1 \wedge \ldots \wedge p_k \wedge p_{k+1}) \equiv \neg p_1 \vee \ldots \vee \neg p_k \vee \neg p_{k+1}.$$

Observe that

$\neg(p_1 \wedge \ldots \wedge p_k \wedge p_{k+1})$

$\equiv \neg((p_1 \wedge \ldots \wedge p_k) \wedge p_{k+1})$ (Gen. Asso. Law)

$\equiv \neg(p_1 \wedge \ldots \wedge p_k) \vee \neg p_{k+1}$ (De Morgan's Law)

$\equiv (\neg p_1 \vee \neg p_2 \vee \ldots \vee \neg p_k) \vee \neg p_{k+1}$ (Ind. Hypothesis)

$\equiv \neg p_1 \vee \neg p_2 \vee \ldots \vee \neg p_k \vee \neg p_{k+1}$

Hence, if P(k) is true, then P(k+1) must also be true. By induction, P(n) must be true for all integer $n \geq 2$.

3. Prove that

$$P(n): 1*2+2*3+...+n(n+1)=\frac{n(n+1)(n+2)}{3}, \forall n \geq 1.$$

Proof: We will prove P(n) by using induction on n, n $\in$ N.

Basis step: When n = 1,

$$\text{RHS}=\frac{1(1+1)(1+2)}{3}=1*2=\text{LHS}.$$

Hence, P(1) is true.

Inductive step: Assuming that P(k) is true, we have

$$1*2+2*3+...+k(k+1)=\frac{k(k+1)(k+2)}{3}, \forall k \geq 1.$$

We must now prove that P(k+1) is also true. Hence, we need to prove that

$$1*2+2*3+...+k(k+1)+(k+1)(k+2)=\frac{(k+1)(k+2)(k+3)}{3}.$$

Observe that

$$1*2+2*3+...+k(k+1)+(k+1)(k+2)$$

$$=[1*2+2*3+...+k(k+1)]+(k+1)(k+2)$$

$$=\frac{k(k+1)(k+2)}{3}+(k+1)(k+2) \qquad \text{(Inductive hypothesis)}$$

$$=\frac{k(k+1)(k+2)+3(k+1)(k+2)}{3}$$

$$=\frac{(k+1)(k+2)(k+3)}{3}.$$

Hence, P(k+1) is true.

By induction, P(n) must be true for all n $\in$ N.

4. Prove that P(n): $\sum_{i=0}^{n} ar^i = \frac{a(r^{n+1}-1)}{r-1}, r \neq 1, n \in N \cup \{0\}$.

Proof: We will prove P(n) by using induction on n, n ∈ N ∪ {0}.

Basis step: When n = 0, $\sum_{i=0}^{0} ar^i = ar^0 = a = \frac{a(r^{0+1}-1)}{r-1}$.

Hence, P(0) is true.

Inductive step: Assuming that P(k), we have $\sum_{i=0}^{k} ar^i = \frac{a(r^{k+1}-1)}{r-1}, r \neq 1, k \geq 0$. We must now prove that P(k+1) is also true. Hence, we need to prove that $\sum_{i=0}^{k+1} ar^i = \frac{a(r^{k+2}-1)}{r-1}, r \neq 1$.

Observe that

$$
\begin{aligned}
&\sum_{i=0}^{k+1} ar^i \\
&= \sum_{i=0}^{k} ar^i + ar^{k+1} \\
&= \frac{a(r^{k+1}-1)}{r-1} + ar^{k+1} \\
&= \frac{a(r^{k+1}-1)}{r-1} + \frac{ar^{k+1}(r-1)}{r-1} \\
&= \frac{a(r^{k+1}-1+r^{k+2}-r^{k+1})}{r-1} \\
&= \frac{a(r^{k+2}-1)}{r-1}, r \neq 1.
\end{aligned}
$$

Hence, P(k+1) is true.

By induction, P(n) must be true for all n∈ N ∪ {0}.

Q: Why does induction work in proving $P(n), n \in S$?

Well-Ordering Principle of Non-negative Integers:
Every non-empty set of non-negative integers must have a least element.

Proving the Validity of Mathematical Induction:
Theorem:
$(P(n_0) \wedge (\forall k \in S, k \geq n_0, P(k) \rightarrow P(k+1)))$
$\rightarrow (\forall n \in S, P(n))$ is a tautology.
Proof: Proof by Contradiction.
Assume that $P(n_0) \wedge (\forall k \in S, P(k) \rightarrow P(k+1))$ is true yet the conclusion $(\forall n \in S, P(n))$ is false to obtain a contradiction. Hence, $\exists n^* \in S$, such that $P(n^*)$ is false. Define a set $F \subset S$ such that $F = \{n \mid n \in S \wedge P(x) = \text{False}\}$. Since $P(n^*)$ is false, $n^* \in F$, implying that $F \neq \emptyset$. If F is a non-empty set of non-negative integers, by the Well-Ordering Principle, F has a least element $m \in F$. Consider the integer $m-1$. Since $m-1 \notin F$, $P(m-1)$ must be true. Recall that n_0 is the least element in S, since $P(n_0)$ is true, $n_0 \notin F$ and $(m-1) \geq n_0$. By assumption, $\forall k \in S, k \geq n_0$, we have $P(k) \rightarrow P(k+1)$. Hence, if $P(m-1)$ is true, then $P(m)$ must also be true, implying that $m \notin F$. This is a contradiction to the assumption that m is the least element in F. Hence, if $P(n_0) \wedge (\forall k \in S, P(k) \rightarrow P(k+1))$ is true, then the conclusion $(\forall n \in S, P(n))$ must also be true.

Warning: Be very careful when using induction.

Example: Prove the proposition P(n): Any n horses must have the same color, $n \geq 1$.
Proof: Let's prove the above assertion using induction on n, $n \geq 1$.
Basis step: Since one horse will always has the same color as itself, P(1) is always true.
Inductive step: Assume that $\forall\ k \geq 1$, P(k) is true. Hence, for any given k horses, they must all have the same color. We must now prove that for any given (k + 1) horses, they must all have the same color. Consider any given (k + 1) horses. Pick an arbitrary horse among these (k + 1) horses, say *Thunderbolt*. By induction hypothesis, without Thunderbolt, the remaining k horses must have the same color. Now, pick another horse, say Lightfoot, among these remaining k horses and compares its color with Thunderbolt. Again, by inductive hypothesis, both Thunderbolt and Lightfoot must have the same color. Hence, Thunderbolt must have the same color as all of these k horses, implying that all these original (k + 1) horses must have the same color!
Hence, by induction, we have proved that any given n horses must have the same color, $n \geq 1$.

Corollary: All humans are men (women) and all humans have the same father (mother).

Q: What's wrong with the above induction proof?

Practice HW: Chpt. 5.1, 1, 3, 5, 7, 9, 19, 21, 31, 41.

In proving some universal quantification, a different but equivalent form of induction, called ***Strong Induction***, can sometimes be applied more easily.

The Second Form of Induction:
Strong InductionTheorem:
If $P(n_0)$ is true, and if $P(n_0+1) \wedge P(n_0+2) \wedge \ldots \wedge P(k) \rightarrow P(k+1)$ is also true, $k \geq n_0$, then $\forall\, n \in S$, $P(n)$ is true. Hence,

$$(P(n_0) \wedge (P(n_0+1) \wedge P(n_0+2) \wedge \ldots \wedge P(k) \rightarrow P(k+1))) \rightarrow (\forall n \in S, P(n)).$$

Proving P(n) using Strong Induction:
Two steps:
1. Basis step:
   Prove that $P(n_0)$ is true.
2. Inductive step:
   $\forall k \geq n_0$, if $P(n_0+1), P(n_0+2), \ldots, P(k)$ are all true, we must prove that $P(k+1)$ is true.

Proving Universal Quantifications using Strong Induction:

Examples:

5. Fundamental Theorem of Arithmetic:

 P(n): $\forall$ n $\in$ N, n > 1, n is either a prime or is a product of primes.

Proof: Let's use strong induction on n, n $\in$ N−{1}.

Basis step: Since $n_0 = 2$ is a prime, P(2) is true.

Inductive step: Assume that P(3), P(4), …, and P(k) are all true, we must prove that P(k+1) is also true.

Hence, if we assume that for all integer k, $2 \le k < n$, k is either a prime or a product of primes, we must prove that k+1 is either a prime or a product of primes.

If k+1 is a prime, then P(k+1) is true and the proof is completed. Otherwise, k+1 must be composite.

If k+1 is composite, $\exists$ x, y $\in$ N such that k+1 = xy, $2 \le x, y < n$. By inductive hypotheses, both x and y must either be a prime or a product of primes, implying that xy must also be a product of primes. Hence, P(k+1) is true if P(3), P(4), …, P(k) are true. By strong induction, $\forall$ n $\in$ N−{1}, n is either a prime or is a product of primes.

6. Prove P(n) using strong induction.
   P(n): Given a function f(n) defined by:

$$f(1) = 1, f(2) = 4,$$
$$f(n) = 2f(n-1) - f(n-2), \forall n \in N, n > 2.$$

For all integers $n \geq 1, f(n) = 3n - 2.$

Proof: Let's use strong induction on n, n $\in$ N.

Basis step:

Observe that there are two base cases in P(n).

When $n = 1, f(1) = 3 * 1 - 2 = 1.$ Hence, P(1) is true.

When $n = 2, f(2) = 3 * 2 - 2 = 4.$ Hence, P(2) is also true.

Inductive step: Assume that P(3), P(4), …, and P(k) are all true, we have $\forall k \in N, 2 \leq k < n, f(k) = 3k - 2$ satisfying the function $f(k) = 2f(k-1) - f(k-2).$

We must now prove that $f(k+1) = 3(k+1) - 2 = 3k + 1$ also satisfy the function $f(k+1) = 2f(k) - f(k-1).$

Observe that

$$\begin{aligned}
&f(k+1) \\
&= 2f(k) - f(k-1) \\
&= 2(3k-2) - [3(k-1) - 2] \\
&= 6k - 4 - 3k + 3 + 2 \\
&= 3k + 1.
\end{aligned}$$

Hence, P(k+1) is true.

By strong induction, P(n) is true for all positive integers n $\in$ N.

Practice HW: Chpt. 5.2, 1, 3, 5.

Q: What if S is not a set of consecutive integers, can we still use the technique of induction?

Given a universal quantification

$P(n)$: $\forall n \in S, P(n)$.

If $S \neq \{n_0, n_0+1, n_0+2, \ldots\}$, induction cannot be applied (directly) to prove the validity of $P(n)$. However, if $P(n)$ can be transformed into another universal quantification

$Q(m)$: $\forall m \in H, Q(m)$,

with $H = \{n_0, n_0+1, n_0+2, \ldots\}$, $n_0 \geq 0$, and $P(n) \leftrightarrow Q(m)$, we can then establish the validity of P by applying induction to $Q(m)$ instead.

Examples:

7. Prove that $P(n)$: For all even integers n, $3n+1$ must be odd.

   Proof: Observe that the domain of disclosure for n is the set of even integers $S = \{2, 4, 6, \ldots\}$, which is not in the right form for an induction proof. However, recall that if n is even, $\exists m \in N$ such that $n = 2m$. We can then use this property to transform the original universal quantification into the following equivalent universal quantification:

   $$Q(m): \forall m \in N, 3(2m)+1 = 6m+1 \text{ is odd.}$$

   Observe that $P(n) \leftrightarrow Q(m)$ and we can prove $P(n)$ by applying induction to $Q(m)$ instead.

   Basis step: When $m = 1$, $6m+1 = 7$, which is odd. Hence, $Q(1)$ is true.

   Inductive step: Assume that $Q(k)$ is true. Hence, $6k+1$ is odd, $k \geq 1$. We must now prove that $Q(k+1)$ is true, or $6(k+1)+1 = 6k+7$ must be odd. Observe that $6k+7 = (6k+1) + 6$. By inductive hypothesis, $6k+1$ is odd. Hence, there exists an integer h such that $6k+1 = 2h+1$ and $Q(k+1) = (6k+1)+6 = (2h+1)+2*3 = 2(h+3)+1$. Since $h+3$ is an integer, $Q(k+1)$ must also be odd. Hence, $Q(k+1)$ is true and, by induction, $\forall m \in N$, $Q(m)$ is true.

   Since $P(n) \leftrightarrow Q(m)$, $P(n)$ must also be true for all even $n \in N$.

8. Let S = {1, 3, 9, …, 3^k, …}.
Define a function $f : S \to N$ such that

$$f(1) = 1,$$

$$f(n) = 3f(\frac{n}{3}) + 1, \forall n \in S - \{1\}.$$

Prove that P(n): $\forall n \in S, f(n) = \frac{3n-1}{2}$.

Proof: Define a function $g : N \cup \{0\} \to N$ such that

$$g(3^0) = 1, \text{ if } m = 0,$$

$$g(3^m) = 3g(3^{m-1}) + 1, m \in N.$$

Consider an equivalent universal quantification

Q(m): $\forall m \in N \cup \{0\}, g(3^m) = \frac{3^{m+1}-1}{2}$.

Observe that P(n) ↔ Q(m) and we can prove P(n) by applying induction to Q(m) instead.

Basis step: If m = 0, $g(3^0) = \frac{3^{0+1}-1}{2} = 1$.

Hence, Q(0) is true.

Inductive step: Assuming that Q(k) is true, we have $g(3^k) = \frac{3^{k+1}-1}{2}$. We must now prove that Q(k+1) is also true. Hence, we need to prove that $g(3^{k+1}) = \frac{3^{k+2}-1}{2}$.

Observe that

$$
\begin{aligned}
&Q(k+1) \\
&= g(3^{k+1}) \\
&= 3g(3^{k})+1 \\
&= 3\left(\frac{3^{k+1}-1}{2}\right)+1 \\
&= \frac{3^{k+2}-3+2}{2} \\
&= \frac{3^{k+2}-1}{2}.
\end{aligned}
$$

Hence, Q(k+1) is true and, by induction, Q(m) is true $\forall m \in N \cup \{0\}$.

Since $P(n) \leftrightarrow Q(m)$, P(n) must also be true for all $n \in S = \{1, 3, 9, \ldots, 3^k, \ldots\}$.

Inductive Approach in Problems Solving:

Two Examples:

1. 2-Max Finding Problem:

Input: An array A[1..n] of distinct integers.
Output: Return the two largest integers in A.

Approach: Sequentially scanning array A to compute the two largest integers.

(1) n = 2:

```
if A[1] > A[2]
   then   largest = A[1];
          s_largest = A[2];
   else   largest = A[2];
          s_largest = A[1];
endif;
```

(2) n > 2: Consider $n = k+1$, $k \geq 2$.

Q: If we have already computed the two largest integers for A[1..k], how do we compute the two largest integers for A[1..k+1] when an extra integer A[k+1] is given?

Q: How do we make use of the information of the current solution to generate a solution to a larger problem?

Observe that

```
if A[k+1] > s_largest
   then  if A[k+1] > largest
            then  s_largest = largest;
                  largest = A[k+1];
            else  s_largest = A[k+1];
         endif;
endif;
```

Algorithm:

```
if A[1] > A[2]             // Initialization
   then largest = A[1];
        s_largest = A[2]
   else  largest = A[2];
        s_largest = A[1]
endif;
for i = 3 to n do          // Extending current solution
   if A[i] > s_largest    // A[i] is one of the two largest integers
      then if A[i] > largest     // A[i] is the current largest integer
             then  s_largest = largest;
                   largest = A[i]
             else   s_largest = A[i]
            endif
   endif
endfor;
```

Q: Do you observe any similarity between this approach and induction?

Comparing Induction and Inductive Algorithm:

Induction	**Inductive Algorithm**
Given proposition P with domain of disclosure S.	Given Problem Π with input I.
Basis Step: Prove for $P(n_0)$, n_0 is smallest integer in S.	Initial Condition(s): Compute $\Pi(I_0)$, I_0 is small enough so that $\Pi(I_0)$ can be computed directly.
Inductive Step: Prove that	Iterative Step: Compute $\Pi(I_1)$ from $\Pi(I_0)$,
$P(n_0+1) \wedge P(n_0+2) \wedge \ldots \wedge P(k)$	$\Pi(I_2)$ from $\Pi(I_0)$ and $\Pi(I_1)$,
$\rightarrow P(k+1)$, $\forall k \geq n_0$.	…, and then $\Pi(I)$, $\lvert I_0 \rvert < \lvert I_1 \rvert < \lvert I_2 \rvert < \ldots < \lvert I \rvert$.

Complexity Analysis:
Basic operations:
Comparisons between elements in A.

Observe that
1. Initialization step requires 1 comparison.
2. For-loop will be executed exactly n-2 times.
3. Either 1, or 2, comparison(s) will be required in executing the for-loop.
4. If A[i] is one of the two largest integers in A[1..i], exactly 2 comparisons will be required.

Best-case Complexity:

$$T_b(n) = 1 + \sum_{i=3}^{n} 1 = n - 1.$$

Worst-case Complexity:

$$T_w(n) = 1 + \sum_{i=3}^{n} 2 = 2n - 3.$$

2. Computing Fibonacci Sequence Number, f_n :

Input: Given a non-negative integer n.

Output: Return the Fibonacci sequence number f_n

Recall that the Fibonacci sequence $\{f_i\}_{i=0}^{\infty}$ can be defined recursively as follows:

$$f_0 = 0,$$
$$f_1 = 1,$$
$$f_n = f_{n-1} + f_{n-2,} \forall n > 1.$$

Given a non-negative integer n, using this recursive definition, we can implement a recursive algorithm to compute f_n

Algorithm 1:

```
int fib1(int n)
   if (n  == 0)
      then return 0;
   if (n  == 1)
      then return 1;
   return fib1(n-1) + fib1(n-2);
end fib1;
```

Let $T(n)$ be the complexity in computing f_n.

Complexity Analysis:

$T(0) = T(1) = constant,$

$T(n) = T(n-1) + T(n-2) + constant, n > 1.$

$$
\begin{aligned}
& T(n) \\
& = T(n-1) + T(n-2) + constant \\
& \le 2T(n-1) + constant \\
& \le 2^2 T(n-2) + constant \\
& \ldots \\
& \le 2^{n-1} T(1) + constant \\
& = O(2^n)
\end{aligned}
$$

Similarly, it can be shown that $T(n) = \Omega(m^n), m > 1.$
Hence, this algorithm is totally impractical!

Q: As in previous example, can we compute f_n based on $f_{n-1}, f_{n-2}, \ldots, f_0$ using inductive algorithm?

Observe that, we can compute f_2 from $f_1 + f_0$, f_3 from $f_2 + f_1$, …, and then f_n from $f_{n-1} + f_{n-2}$.
Hence, we can compute $f_2, f_3, \ldots, f_{n-1}, f_n$, in this order, simply by adding the previous two Fibonacci sequence numbers that have already been computed.

Algorithm 2:

```
int fib2(int n)
    if (n  == 0)
        then return 0;
    if (n  == 1)
        then return 1;
    first = 0;        //computing f_2, f_3, ..., f_{n-1}, f_n sequentially
    second = 1;
    for k = 2 to n do
        next = first + second;
        first = second;
        second = next;
    endfor;
    return next;
end fib2;
```

Complexity Analysis:

$$T(n) = \sum_{i=2}^{n} C = \Theta(n).$$

Q: Can we still compute f_n faster?

Yes. Using fast matrix multiplications algorithm, $T(n) = \Theta(\lg n)$.

Observations:

1. Some mathematical structures and functions can sometimes be computed using information of some previously computed but simpler structures.
2. The design and execution of an inductive algorithm are similar to a typical induction proof.
3. In executing an inductive algorithm, structures are being computed in a linear fashion without recursion so as to eliminate any re-computations.
4. In order to trigger this computation process, we must have one or more basic structures that require only simple computations.
5. In order to extend this computational process, there must be a relation that relates the given basic structures, together with some previously computed structures, with the more complicated structures to be computed.