6.2 Recursion

Notebook:	Discrete Mathematics [CM1020]		
Created:	2019-10-07 2:31 PM	**Updated:**	2019-12-07 12:33 PM
Author:	SUKHJIT MANN		

Cornell Notes	**Topic:** 6.2 Recursion	Course: BSc Computer Science Class: Discrete Mathematics-Lecture Date: December 06, 2019

Essential Question:

What are recursion, recurrence, and recurrence relations?

Questions/Cues:

- What is recursion?
- What is a Recursively defined function?
- What is a recursively defined set?
- What is recursive algorithm?
- What is a recurrence relation?
- What is a linear recurrence?
- What is a arithmetic sequence?
- What is a geometric sequence?
- What is a divide and conquer recurrence?
- How to solve recurrence relations?
- How do we use induction when solving a recurrence relation?

Notes

Definition

- Sometimes it is **difficult** to define a mathematical **object** (e.g. a function, sequence or set) explicitly; it is easier to define the object **in terms of the object itself**
- This process is called **recursion.**

Recursively defined functions

A recursively defined function f with domain ℕ is a function defined by:

- BASIS STEP: specify an initial value of the function
- RECURSIVE STEP: give a rule for finding the value of the function at an integer from its values at smaller integers

- Such a definition is called a **recursive** or **inductive definition**
- Defining a function f (n) from the set ℕ to the set ℝ is the same as a **sequence** $a_0, a_1 ...$ where $\forall i \in \mathbb{N}, a_i \in \mathbb{R}$.

Examples

Let's give a recursive definition of the sequence $\{a_n\}$, n = 1, 2, 3, ..., in the following cases:

1. $a_n = 4n$
2. $a_n = 4^n$
3. $a_n = 4$

- There may be more than one correct answer to each sequence
  1. As each term in the sequence is greater than the previous term by 4, this sequence can be defined by setting $a_1 = 4$ and declaring that $\forall n \geq 1$ $a_{n+1} = 4 + a_n$
  2. As each term is 4 times its predecessor, this sequence can be defined as $a_1 = 4$ and $\forall n \geq 1$ $a_{n+1} = 4a_n$
  3. This sequence can be defined as $a_1 = 4$ and $\forall n \geq 1$ $a_{n+1} = a_n$.

Recursively defined sets

Sets can also be defined recursively, by defining two steps:

- BASIS STEP: where we specify some initial elements
- RECURSIVE STEP: where we provide a rule for constructing new elements from those we already have.

Example:

- Consider the subset S of the set of integers recursively defined by:
  1. BASIS STEP: $4 \in S$
  2. RECURSIVE STEP: if $x \in S$ and $y \in S$, then $x + y \in S$
- Later we will see how it can be proved that the set S is the set of all positive integers that are multiples of 4.

Recursive algorithms

Definition 1:

- An algorithm is a finite sequence of precise instructions for performing a computation or for solving a problem.

Definition 2:

- An algorithm is called recursive if it solves a problem by reducing it to an instance of the same problem with a smaller input.

Example

Let's give a recursive algorithm for computing n!, where n is a nonnegative integer:

- n! can be recursively defined by the following two steps:
  BASIS STEP: 0! = 1
  RECURSIVE STEP: n! = n (n - 1)! when n is a positive integer

- The pseudocode of this algorithms can be formalised as:

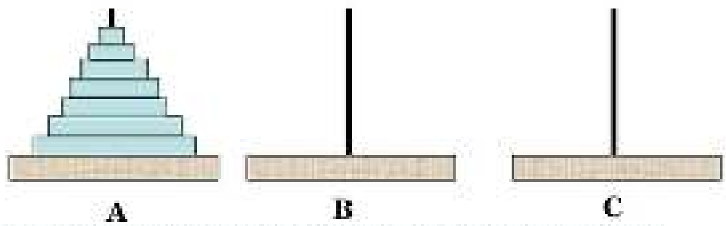
```
procedure factorial(n: nonnegative integer){
        if n = 0 then return 1
        else
        return n factorial (n - 1)
}
```

Definitions

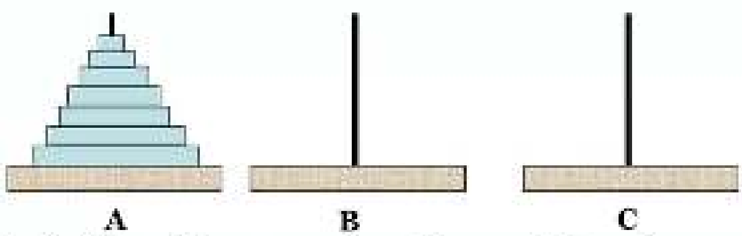
- A recurrence relation is an equation that defines a sequence based on a rule that gives the next term as a function of the previous term
- An infinite **sequence** is a **function** from the set of positive integers to the set of real numbers
- In many cases, it can be very useful to **formalise** the problem as a sequence before solving it.

Example: Hanoi Tower

- The game of Hanoi Tower is played with a set of discs of graduated size and a playing board consisting of three spokes for holding the discs
- The object of the game is to transfer all the discs from spoke A to spoke C by moving one disk at a time without placing a larger disc on top of a smaller one.

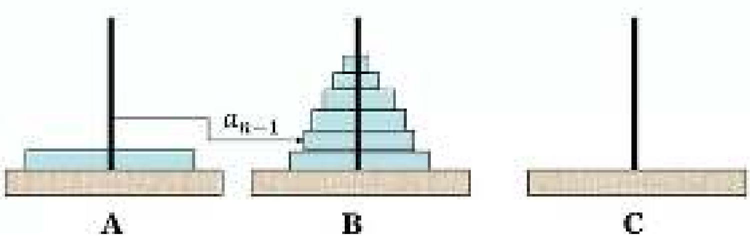


Let a_n be the minimum number of moves to transfer n discs from one spoke to another:



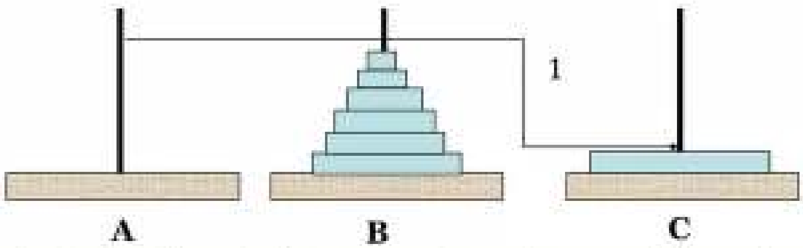
Let a_n be the minimum number of moves to transfer n discs from one spoke to another:

- in order to move n discs from A to C, we must move the first n−1 discs from A to B by a_{n-1} moves



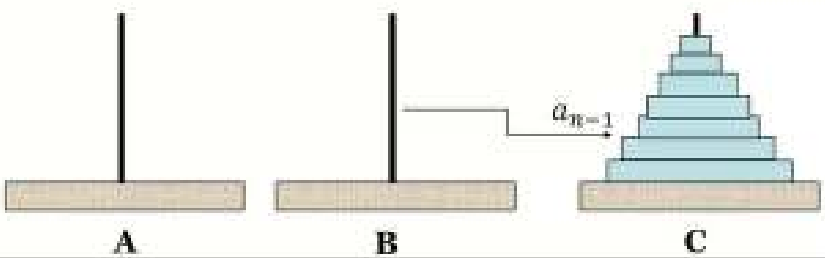
Let a_n be the minimum number of moves to transfer n discs from one spoke to another:

- in order to move n discs from A to C, we must move the first n-1 discs from A to B by $\boldsymbol{a_{n-1}}$moves
- then, move the last (and also the largest) disc from A to C by **one** move



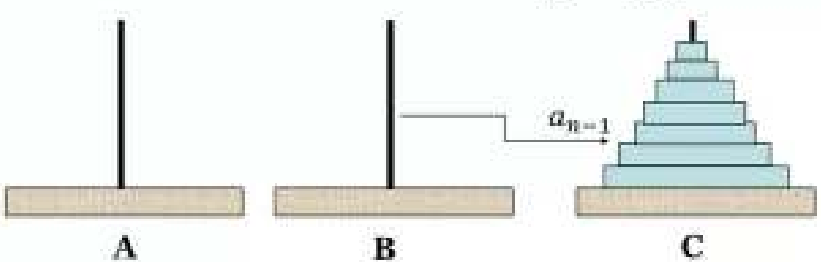
Let a_n be the minimum number of moves to transfer n discs from one spoke to another:

- in order to move n discs from A to C, we must move the first n-1 discs from A to B by $\boldsymbol{a_{n-1}}$moves
- then, move the last (and also the largest) disc from A to C by **one** move
- then, remove the n - 1 discs again from B to C by $\boldsymbol{a_{n-1}}$moves



Let a_n be the minimum number of moves to transfer n discs from one spoke to another:

- in order to move n discs from A to C, we must move the first n-1 discs from A to B by $\boldsymbol{a_{n-1}}$moves
- then, move the last (and also the largest) disc from A to C by **one** move
- then, remove the n - 1 discs again from B to C by $\boldsymbol{a_{n-1}}$moves
- thus, the total number of moves is: $\boldsymbol{a_n = 2a_{n-1} + 1}$.



Linear recurrences

A linear recurrence is a relation in which each term of a sequence is a linear function of earlier terms in the sequence.

- There are two types of linear recurrence:
  - linear **homogeneous** recurrences:
    - formalised as $a_n = c_1 a_{n-1} + c_2 a_{n-2} + \dots + c_k a_{n-k}$
    - where $c_1, c_2, \dots, c_k \in \mathbb{R}$, and k is the degree of the relation
  - linear **non-homogeneous** recurrences:
    - formalised as $a_n = c_1 a_{n-1} + c_2 a_{n-2} + \dots + c_k a_{n-k} +$ f(n)
    - where $c_1, c_2, \dots, c_k \in \mathbb{R}$, f(n) is a function depending only on n, and k is the degree of the relation.

Example: first order recurrence

Let's consider the following case :
- a country with currently **50 million** people that:
  - has a population growth rate (birth rate minus death rate) of **1%** per year
  - receives **50,000** immigrants per year
- **question**: find this country's population in **10 years** from now.

- This case can be modelled as the following **first-order linear recurrence**:
  - where a_n is the population in n years from now
  - $\forall n \in \mathbb{N}, a_{n+1}$ is expressed as $a_{n+1} = 1.01\, a_n + 50{,}000$
  - $a_0 = 50{,}000{,}000$.

Example: second order recurrence

Let's consider the following sequence:
- 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, ...
- where each number is found by adding up the two numbers before it

This sequence can be modelled as the following second-order linear recurrence:
- $a_n = a_{n-1} + a_{n-2}$
- $a_0 = 0$
- $a_1 = 1$

This sequence is famously known as the **Fibonacci sequence**.

Arithmetic sequences

- A sequence is called ***arithmetic*** if the difference between consecutive terms is a constant c
- $\forall n, a_{n+1}$ is expressed as $a_{n+1} = a_n + c$ and a_n =a.

Example:

- The sequence 2, 5, 8, 11, 14, ... is **arithmetic** with an initial term of $a_0 = 2$ and a common difference of 3
- 30, 25, 20, 15,... is arithmetic with an initial term of $a_0 = 30$ and a common difference of -5.

Geometric sequences

- A sequence is called ***geometric*** if the ratio between consecutive terms is a constant r
- $\forall n, a_{n+1}$ is expressed as $a_{n+1} = r\, a_n$ and $a_0 = a$.

Example:

- The sequence 3, 6, 12, 24, 48, ... is geometric with an initial term of $a_0 = 3$ and a common ratio of 2
- 125, 25, 5, 1, 1/5, 1/25, ... is geometric with an initial term of $a_0 = 125$ and a common ratio of 1/5.

Divide and conquer recurrence

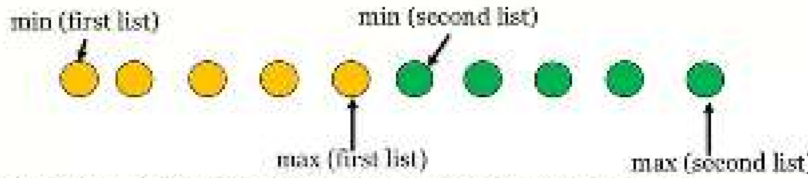
A divide and conquer algorithm consists of three steps:

- dividing a problem into smaller subproblems
- solving (recursively) each subproblem
- and then combining solutions to find a solution to the original problem.

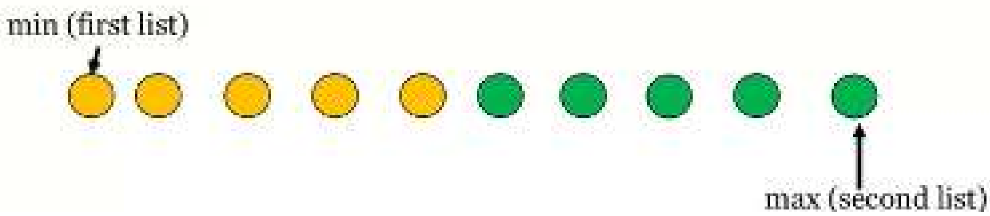
Example

Let's consider the problem of finding the minimum of a sequence $\{a_n\}$ where $n \in \mathbb{N}$

- if n=1, the number is itself min or max
- if n>1, divide the numbers into two lists
- order the sequence
- find the min and max in the first list
- then find the min and max in the second list



- then infer the min and max of the entire list.



Solving linear recurrence

- Let $a_n = c_1\, a_{n-1} + c_2\, a_{n-2} + \ldots + c_k\, a_{n-k}$ be a linear homogeneous recurrence

- If a combination of the geometric sequence $a_n = r^n$ is a solution to this recurrence, it satisfies $r^n = c_1 r^{n-1} + c_2 r^{n-2} \ldots c_k\, r^{n-k}$

- By dividing both sides by r^{n-k}, we get: $r^k = c_1 r^{k-1} + c_2 r^{k-2} \ldots c_k$

This equation is called the **characteristic equation**.

Solving linear recurrence

Solving this equation is the first step towards finding a solution to linear homogeneous recurrence:

- If r is a solution of the equation with multiplicity p, then the combination $(\alpha + \beta n + \gamma n^2 + \ldots + \mu n^{p-1})r^n$ satisfies the recurrence
- We will examine some examples of how this works in the next section.

Example: solving Fibonacci

Let's consider solving the Fibonacci recurrence relation:

$f_n = f_{n-1} + f_{n-2}$, with $f_0 = 0$ and $f_1 = 1$

Solution:

- The characteristic equation of the Fibonacci recurrence relation is:
  - $r^2 - r - 1 = 0$
- It has two distinct roots, of multiplicity 1:
  - $r_1 = (1+\sqrt{5})/2$ and $r_2 = (1 - \sqrt{5})/2$
- So, $f_n = \alpha_1 r_1^n + \alpha_2 r_2^n$ is a solution

Example: solving Fibonacci

To find α_1 and α_2 we need to use the initial conditions.

- From:
  - $f_0 = \alpha_1 + \alpha_2 = 0$
  - $f_1 = \alpha_1(1+\sqrt{5})/2 + \alpha_2(1+\sqrt{5})/2 = 1$
- we can find $\alpha_1 = 1/\sqrt{5}$ and $\alpha_2 = -1/\sqrt{5}$

- The solution is then formalised as:

  $f_n = 1/\sqrt{5} .((1+\sqrt{5})/2)^n)) - 1/\sqrt{5} .((1-\sqrt{5})/2)^n))$

Example 2

Let's consider another sequence:

- $a_n = -3a_{n-1} - 3a_{n-2} - a_{n-3}$
- a_0 =1, $a_1 = -2$ and $a_2 = -1$

Solution:

- The characteristic equation of this relation is: $r^3 + 3r^2 + 3r + 1 = 0$
- It has one distinct root, whose multiplicity is 3
  $r_1 = -1$
- So, $a_n = (\alpha_0 + \alpha_1 n + \alpha_2 n^2) r_1^n$ is a solution.

Example 2

- To find α_0, α_1 and α_2 we need to use the initial conditions.
  - From:
    - $a_0 = \alpha_0 = 1$
    - $a_1 = -(\alpha_0 + \alpha_1 + \alpha_2) = -2$
    - $a_2 = -(\alpha_0 + 2\alpha_1 + 4\alpha_2) = -1$
  - we can find $\alpha_1 = 1, \alpha_2 = -3$ and $\alpha_3 = -2$

- The solution is then formalised as:
  $a_n = (1 + 3n - 2n^2)(-1)^n$

Induction for solving recurrence

Sometimes, it is easier to verify a recurrence relation solution using strong induction.

Example:

- Let's try to prove the following:
  - P(n): the sequence $f_n = 1/\sqrt{5}(r_1^n - r_2^n)$ verifies the Fibonacci recurrence, where:
    - $r_1 = (1+\sqrt{5})/2$
    - $r_2 = (1-\sqrt{5})/2$ are the roots of $r^2 - r - 1 = 0$

Induction for solving recurrence

- Let's verify P(2):
  - $f_1 + f_0 = 1/\sqrt{5}(r_1 - r_2) = 1/\sqrt{5}(\sqrt{5}) = 1 = f_2$
  - because $f_2 = 1/\sqrt{5}(r_1^2 - r_2^2) = 1$
  - which verifies the initial condition.

- Let k ∈ ℕ, where P(2) P(3)...P(k) are all true
- Let's verify P(k+1):
  - $f_n + f_{n-1} = \frac{(r_1^n - r_2^n)}{\sqrt{5}} + \frac{(r_1^{n-1} - r_2^{n-1})}{\sqrt{5}} = r_1^{n-1}(r_1 + 1)/\sqrt{5} + r_2^{n-1}(r_2 + 1)/\sqrt{5} = r_1^{n-1} * r_1^2 + r_2^{n-1} * r_2^2 = r_1^{n+1} + r_2^{n+1}$ which equals f_{n+1}

We conclude that P(k+1) is true and the strong induction is verified.

Summary

In this week, we learned what recursion is, what is recursively-defined set & function are and what a recursive algorithm is. Also we explored what a recurrence relation is, the difference types of sequences, what linear & divide and conquer recurrences are and lastly how to solve recurrence relations.