C#程序设计及应用

唐大仕

dstang2000@263.net

北京大学

第5章 基础类及常用算法

唐大仕
dstang2000@263.net
http://www.dstang.com

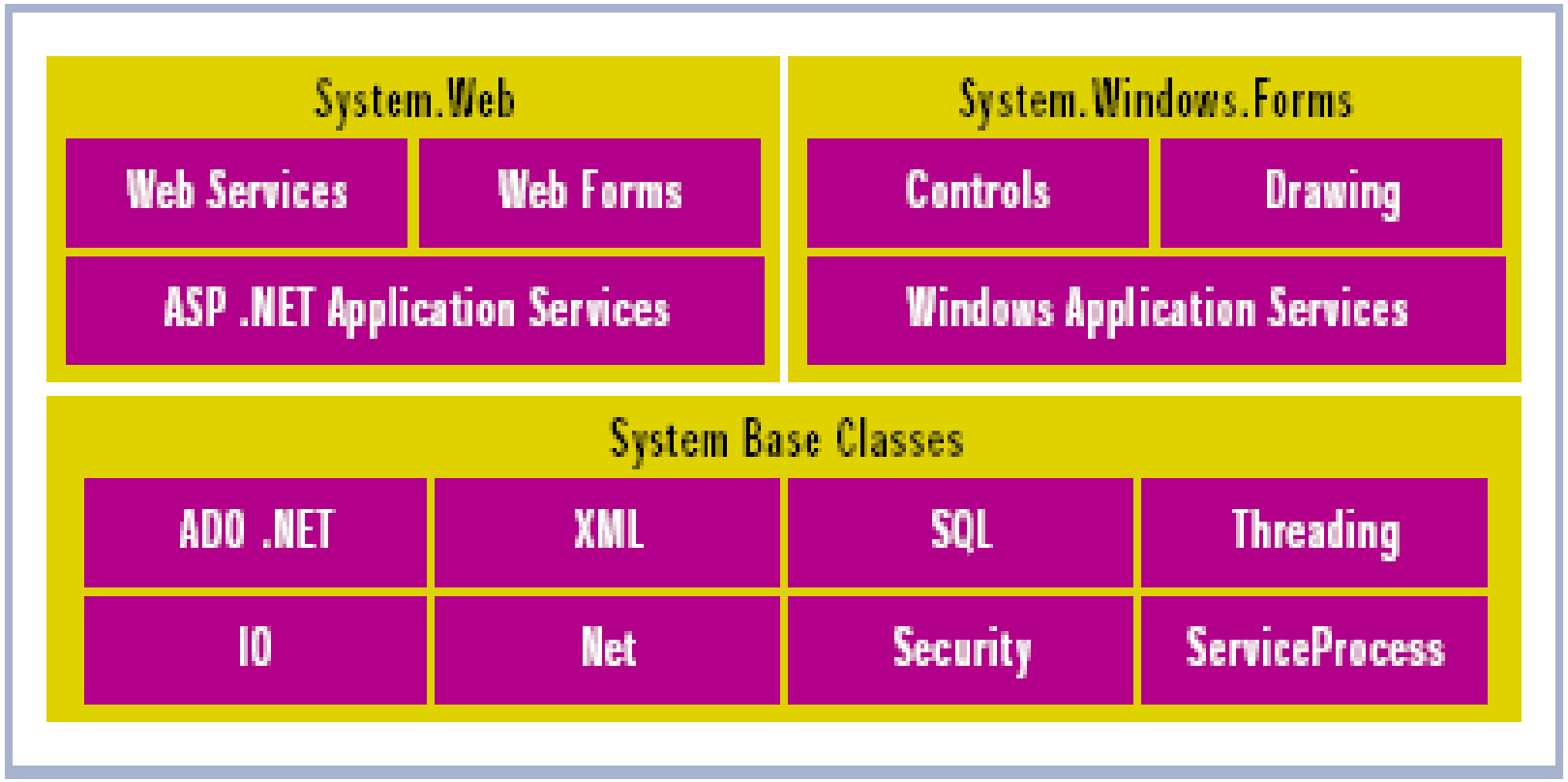
本章内容

- 1 DotNet基本类库
- 2 类型转换
- 3 数学、文字、日期
- 4 数组、集合、泛型
- 5 常用算法
- 6 程序的调试

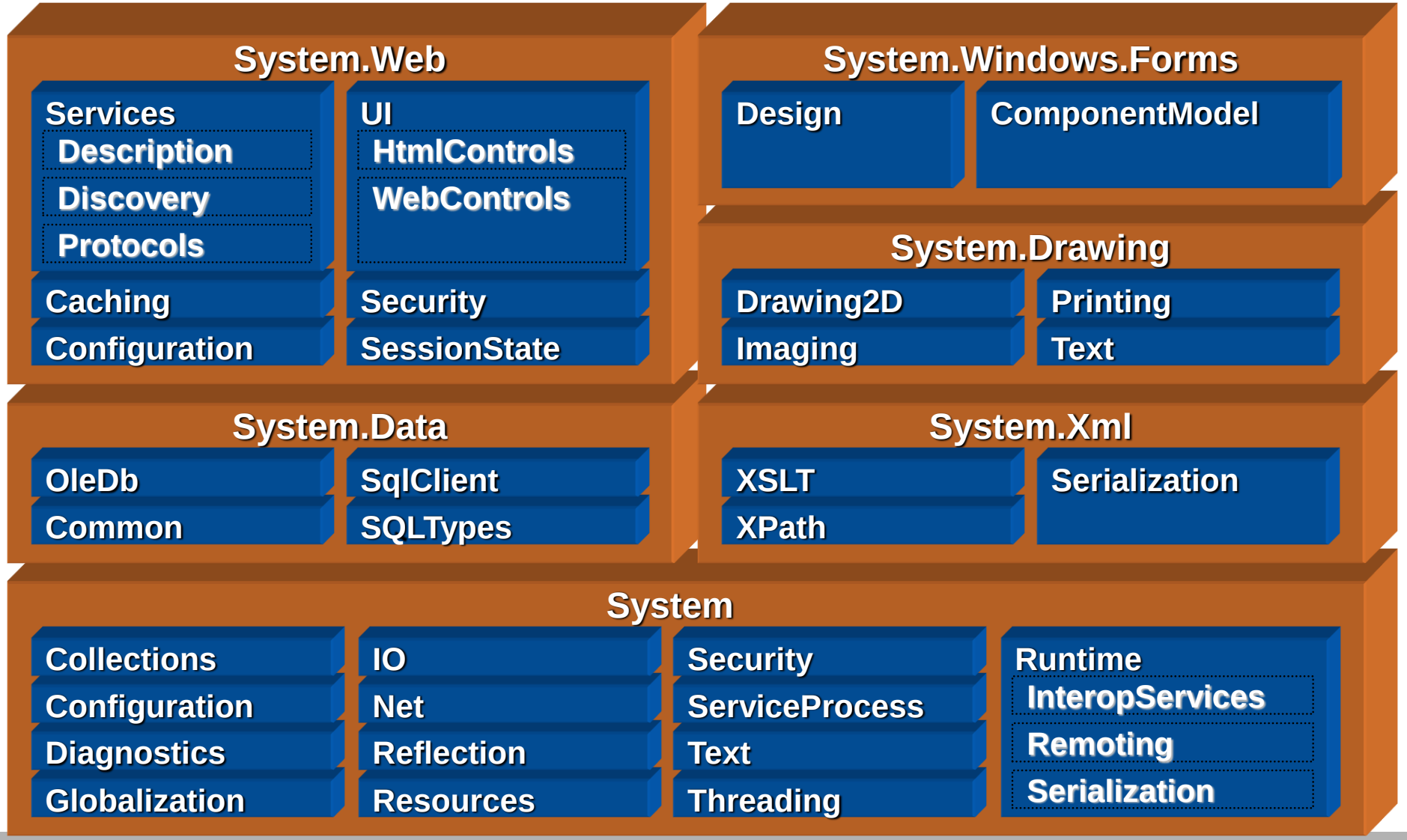
5.1 DotNet基本类库

1 DotNet基本类库

统一的编程API：.NET Framework 类库



统一的编程API：.NET Framework 类库



5.2 基本类型及类型转换

2 基本类型及类型转换

任何事物都是对象

- 任何事物都是object类的子类
  - 一个函数如果需要object参数，则可以代入任意参数
  - 任何对象都有以下方法
    - ToString()　Equals()　GetType()
    - MemberwiseClone()等
  - 常量也是对象
    - 3.ToString()
    - “Hello”.Length

表达式中的类型转换

- 当有不同种类的混合运算时:
- int→long→float→double
- (所有的byte, short, char 等转为int)

强制类型转换

- 在表达式前面用（类型）来表示
- double d=3.14;
- int a = (int) d;
- float b = (float)(d+1.5);
  - 注意圆括号

类型转换函数

- System.Convert类有以下static方法
  - ToDouble(…)
  - ToInt32(…)
  - ToDateTime(…)

基本类型

- 关键字含有等价的类
  - int 即 System.Int32
- 含有一些特殊的属性或方法
  - int.MaxValue
  - double.NaN
  - Double.PositiveInfinity
  - Double.IsNaN(…)

数与字符串的转换

- int.Parse(string)
  - int ans;
  - int.TryParse(textBox1.Text,out ans)
- double.ToString();
  - ToString("#0.00")

5.3 几个常用类

3 几个常用类

Math类

- 提供了相关的数学方法
  - Abs()
  - Sin() Cos() Tan()
  - Round()
  - Exp() Log()
  - Pow() 乘方

Random类

- .Next(100)　　0到100之间(不含100)
- .NextDouble()　0到1之间
- Random得到的是伪随机数
  - 如果要用更强的随机数，可以使用
    - System.Security.Cryptography
    - .RNGCryptoServiceProvider

DateTime及TimeSpan

- DateTime 是值类型
  - new DateTime(y,m,d,h,m,s)
  - .Now
  - .ToString("yyyy-MM-dd HH:mm:ss“)
  - .AddMinutes(5)
  - .Year, .Month, .Day, .Date
- TimeSpan
  - 两个日期相减，可以得到一个TimeSpan

String类

- == + []
- .Length .IndexOf .LastIndexOf
- .StartsWith, .EndsWith
- .Substring(idx, len) 注意第二个参数
- .Trim, .TrimEnd, .PadLeft, .Insert, .Remove
- .Split(';'), string.Join

String 及 StringBuilder

- String内容不可变（immutable）
- StringBuilder内容可变
  - .Append，.Remove, .Replace
  - .Length，.ToString
- 在循环体中用 s+=….可能会带来效率问题

5.4 数组与集合

4 数组与集合

数组

- 声明
  - int [] a;
  - int [,] b;
- 分配空间
  - a = new int[5];
  - b = new int[4,5];

集合类

- System.Collections命名空间
- 数组列表 ArrayList
  - 相当于动态数组,实现IList
- 哈希表 Hashtable
  - 相当于键/值的集合, 实现 IDictionary
  - 用[]进行访问，表示获取、增加、删除、修改
  - 提示：用于查询时，比线性搜索的效率要高，可用于程序的优化
- 栈和队列 Stack Queue

使用foreach访问数组及集合

- foreach(类型 变量 in xxxx)
- 其中xxxx必须是实现了实现 IEnumerable 接口或含有 GetEnumerator 方法的类型
- 这个方法的原型是 IEnumerator **GetEnumerator();**
- 返回的是一个接口 IEnumerator
  - Current属性
  - MoveNext 及 Reset 方法

泛型集合

- 泛型具有更好的类型检查及性能
- System.Collections.Generic名称空间
  - 列表　List / LinkedList　　SortedList
  - 字典　Dictionary　　SortedDictionary
  - 集　　HashSet　　SortedSet
  - 栈,队列　Stack, Queue

示例

- ListTest
  - List的Add方法， Count属性， [i] 索引器
  - foreach的遍历
- HashtableTest
  - Hashtable的 [ ] 索引，可以表示获取/加入/修改/删除(置为null)
  - Foreach 遍历

排序

- 自己写排序程序
- 使用SortedXXXX类
- 使用Array.Sort方法
  - 示例ArraySort.cs
  - Array.Sort(ary, (a,b)=>a.Length-b.Length)

Data Type			Inherits from		
Array	Index Based	Generic Fixed Length	IEnumerable<T> Icomparable<T> Iequatable<T>	String[] strarr=new string[10]; It is strongly data type with fixed sized, so it is fast	makes limitation by fixed size
ArrayList	Index Based	Non Generic	Ilist Icollection IEnumerable	Arraylist objarr=new Arraylist No Data Type + No Dimension	Difficult to find item just via Index ,No Key
HashTable	Key Value Pair	Non Generic	Idictionary Icollection IEnumerable Iserializable	Determine an index for each item Hashtable objhash=new Hashatable(); Objhash.add(1001,"Mahsa")	Arraylist is faster due to hashtable must do key conversion and it consume time
IDictionary<T>	Key Value Pair	Generic	Icollection IEnumerable	Similar to Hashtable but it is generic	
List<T>	Index Based	Generic	Ilist<T> Icollection<T> IEnumerable<T>	Like Array is strong type Like ArrayList has No Dimension Modify(Add,Remove)	
Ilist<T>	Index Based	Generic	Icollection<T> IEnumerable<T>	Modify(Add,Remove) ***Interface helps to future changes***	Ilist can find the indexof item , **IList[i] √**
ICollection<T>	Randomly	Generic	IEnumerable<T>	Modify(Add,Remove), countable	Randomly,~~**ICollection[i]**~~
IEnumerable<T>	Index Based	Generic	IEnumerable	IEnumerable is read only , suitable to iterate through collection and cannot modify	bring all data from server to client and then filter them
IQueryable<T>	Index Based	Generic	IEnumerable<T>	Iqueryable is suitable for runtime query and reduce overhead from memory, improve performance	Iqueryable bring filtered data from server to client NOT all of them
Stack	Prioritized		Icollection IEnumerable	Non Generic Stack: System.Collections.Stack Generic Stack: System.Collections.Generic.Stack<int>	
Queue	Prioritized		Icollection IEnumerable	Non Generic Queue: System.Collections.Queue Generic Queue: System.Collections.Generic.Queue<int>	

5.5 常用算法

5 常用算法

算法

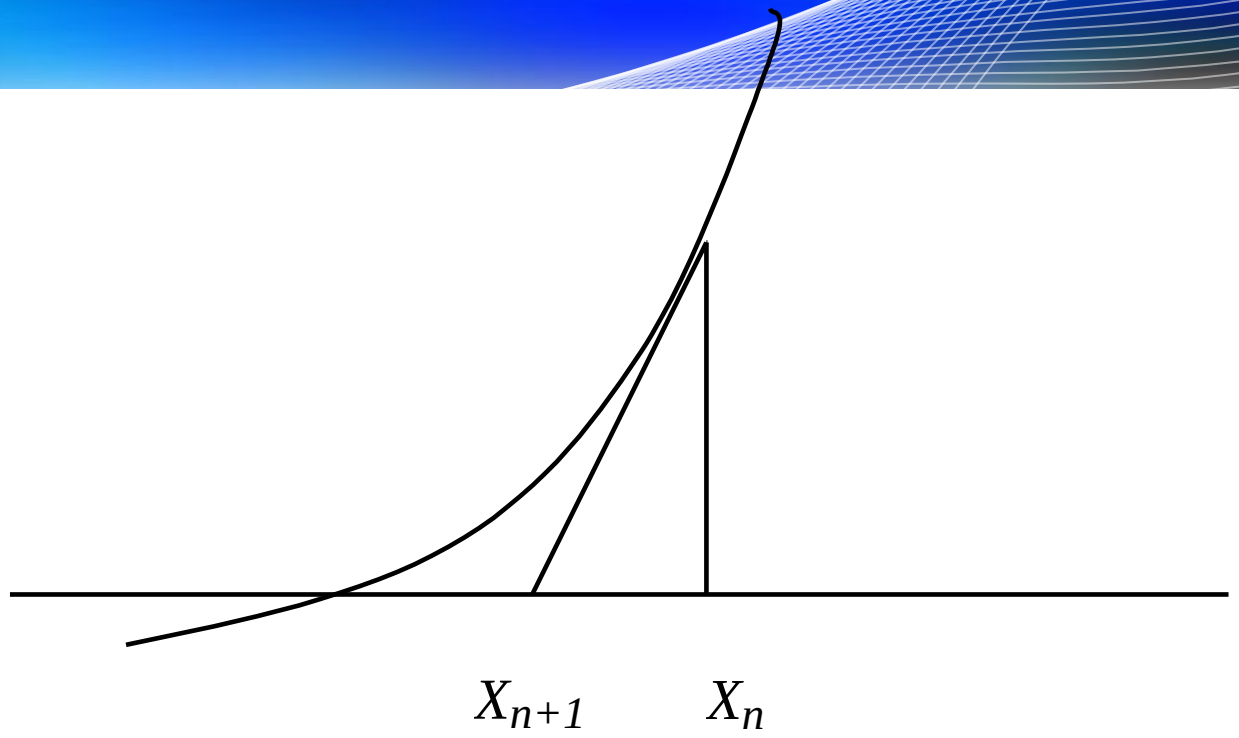
- 指令的有限序列
- 特点：
  - 有穷性
  - 确定性
  - 可行性
  - 输入、输出

遍试

- 逻辑上：针对所有可能的情况进行判断
- 形式上： For 中用 If
  - 示例： 韩信点兵
  - 水仙花数 $1^3+5^3+3^3=153$
  - 完全数 $28=1+2+4+7+14$
  - 相亲数
  - 验证哥德巴赫猜想
  - 其他：百鸡问题,鸡兔同笼问题,百分币,佩尔方程

迭代

- 逻辑上：多次使用同一算法
- 形式上：while中用a = f(a)
  - 示例： 求平方根
  - 倍边法求Pi
  -
  - 其他，如：数字平方和、 Mandelbrot集,Julia集

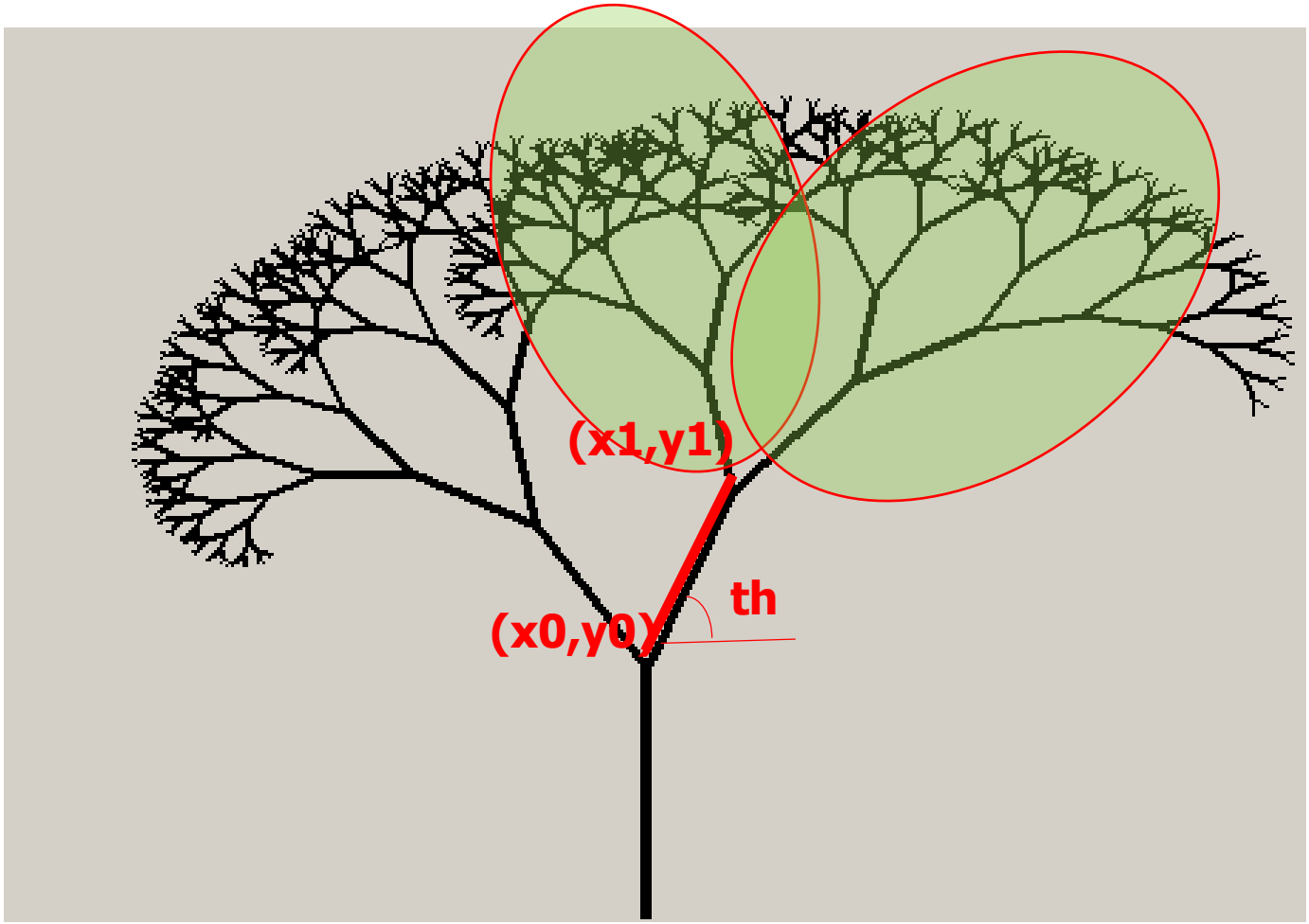


$$x_{n+1} = x_n - f(x_n) / f'(x_n)$$

递归

- 逻辑上：一个问题化为同样的问题
- 形式上：自己调用自己
  - 示例：1.求阶乘
  - 2.菲波那契数列
  - 3. Cayley树
  - 其他，Koch分形集

递归算法的基本思想：分而治之



更多递归现象

小结

- 遍试： for中用if
- 迭代: while中a=f(a)
- 递归: f(n)中用f(n-1)