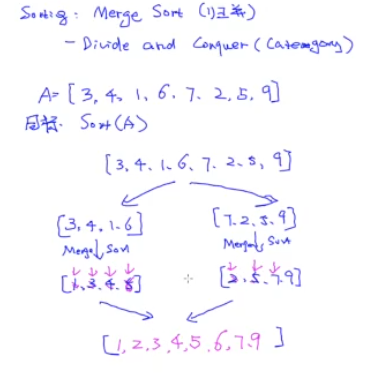
**1. merge sort归并排序**



图片包含 文字

描述已自动生成

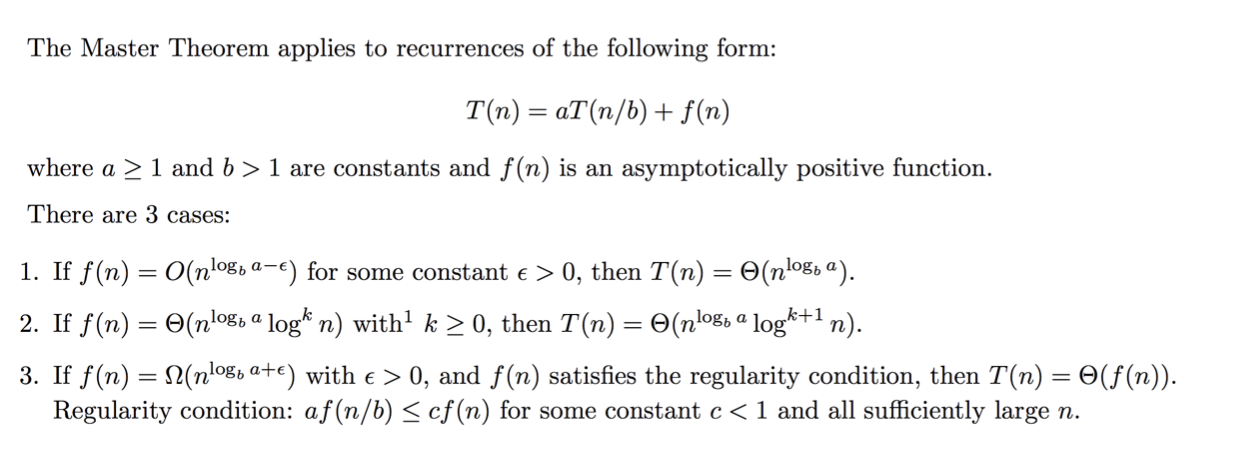
图片包含 文字

描述已自动生成

T(n) = O(n\*logn)

**2. Master Theorem**

一个问题写成比它更简单的小问题



时间复杂度<https://www.jianshu.com/p/f4cca5ce055a>

Master <http://people.csail.mit.edu/thies/6.046-web/master.pdf>

**3. Fibonanci Number**

见jupter

时间复杂度：

T(n) = T(n-2) + T(n-1)

T(n) = 2^n

图片包含 文字, 白板

描述已自动生成

空间复杂度（针对一些递归算法）

上下文切换：变量、stack（操作系统知识）

Fib8🡪fib7就是一次上下文切换，stack会多一层

图片包含 文字

描述已自动生成

到fib2会出栈，因为我们知道它的数值是多少。

求fib3需要知道fib2和fib1，而两者已知（=1），所以可以求得fib3=2，这样fib3就可以出栈。

求fib4需要知道fib3和fib2，对于fib2进栈，但我们知道fib2=1，所以用1来替代fib2，然后fib2出栈。

从fib4到fib5，此时求解fib5需要知道fib3，fib3需要知道fib1和fib2，然后fib1和fib2分别依次进栈出栈，求得fib3。Fib3再与fib4一起求得fib5.

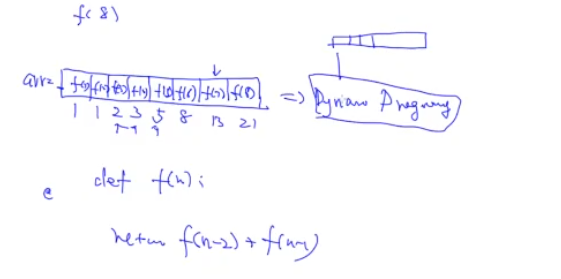
直到求解完fib8，它出栈，栈内只剩main函数。

所以求解fib8时，我们最多用到8个内存空间。

空间复杂度O(n).

**4. dynamic program**

能复用的变量、资源等等尽量复用



用动态规划的思想改进算法，见jupter 2

因为求解某个fib数值时，它只依赖前两个数值，所以我们只需要维护前两个空间。

<https://algorithms.tutorialhorizon.com/dynamic-programming-edit-distance-problem/>

**5. P, NP, NP hard, NP complete**

图片包含 文字

描述已自动生成

图片包含 人员, 文字

描述已自动生成

量子计算

图片包含 文字, 白板

描述已自动生成