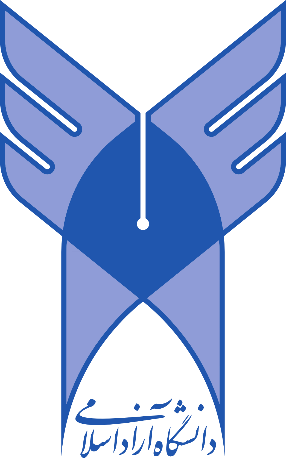
**C:\Program Files (x86)\Hamoon Soft\Chalipa\Chalipa Clipart\Brdr01.wmf**

دانشگاه آزاد اسلامی تهران مرک

مجتمع آیت الله هاشمی رفسنجانی

عنوان :

**تحلیل پیچیدگی الگوریتم ها**

رشته تحصیلی :

**مهندسی کامپیوتر نرم افزار**

استاد راهنما:

**سرکارخانم دکتر نیره زاغری**

تهیه کنندگان:

**حسن پرسته**

**مجتبی سیانکی**

**زمستان 99**

**C:\Program Files (x86)\Hamoon Soft\Chalipa\Chalipa Clipart\Brdr01.wmf**

**پیچیدگی زمانی:**

از نماد O بزرگ برای توصیف ماهیت مجانبی یک سیستم در زمان رشد ورودی استفاده می‌شود. به طور معمول ما از این نماد برای توصیف زمان اجرای یک الگوریتم بهره می‌گیریم؛ اما از آن برای توصیف پیچیدگی فضایی و یا حتی سیستم‌های خارج از حوزه علوم کامپیوتر نیز می‌توان بهره گرفت. در این نوشته فرض می‌کنیم که این نماد برای توصیف پیچیدگی زمانی یک الگوریتم استفاده می‌شود؛ مگر این که صراحتاً به چیز دیگری اشاره کنیم.

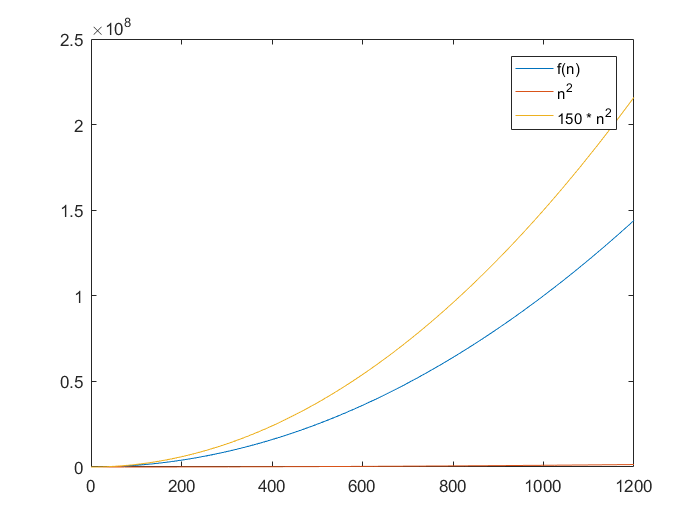
**ϴ بزرگ (تتا)**

به طور رسمی برای اینکه ϴ(g(n) به توصیف تابع f(n) بپردازد، باید ثابت‌های c1 ،c2 و n\_o مثبتی وجود داشته باشند که رابطه زیر در مورد آن‌ها برقرار باشد:

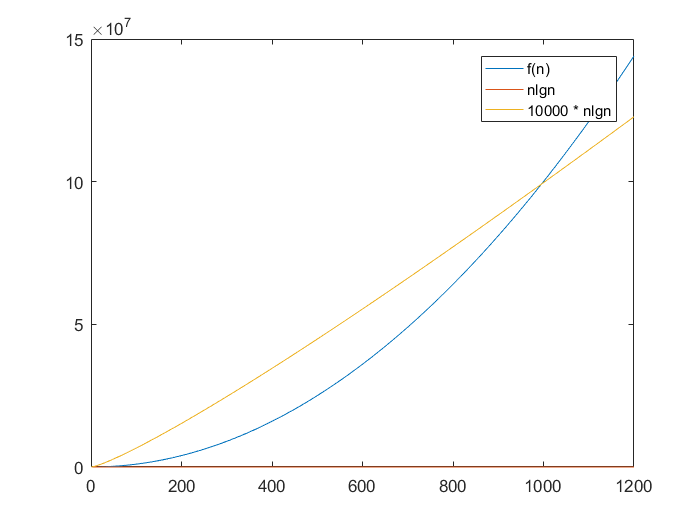
0 <= c1 \* g(n) <= f(n) <= c2 \* g(n) for all n >= n\_o

برای روشن‌تر شدن موضوع تصور کنید الگوریتمی دارید که به وسیله یک معادله پیچیده (f(n توصیف شده است. شما می‌توانید (f(n را از طریق معادله بسیار ساده (g(n که همه مقادیر ثابت، ضریب و جمله‌های با مرتبه پایین آن حذف شده‌اند، با نمادگذاری تتای بزرگ توصیف کنید، اگر و تنها اگر دو ثابت دلخواه به (g(n اضافه کنید که بتوانند تابع اصلی (f(n شما را که از یک اندازه داده ورودی خاص n\_o عبور می‌کند، «ساندویچ» کنند.

مثال: فرض کنید (f(n) = 100\*n^2 + 4\*n\*lg(n باشد. ما می‌توانیم نشان دهیم که n^2 می‌تواند این تعریف را با معادله‌ها و گراف زیر تأمین کند:



اینک اگر تلاش کنیم تابع (f(n را با nlgn، ساندویچ کنیم، می‌بینیم که کران بالا با رشد n به طور ناگزیر از (f(n عبور می‌کند و مهم نیست که مقدار ثابت را چه مقدار تعیین کرده باشیم:



بنابراین از نظر فنی، هر بار که یک تابع با O بزرگ فوق را فراخوانی کنیم، باید به جای (O(n آن را (ϴ(n بنامیم.