

## Algorithms بالعربي

إيماننا بأن أحد أهم أسباب تأخر الأمة الإسلامية هو ضعفها العلمي , وأن أحد أهم أسباب ضعفها العلمي هو قلة المحتوى العلمي العربي مقابل وفرة المحتوى الإنجليزي في كل العلوم , وأنه لا يمكن أن يوجد محتوى عربي قوي يناقش المحتوى الإنجليزي المهور الموجود حاليا إلا بعد سنين من المحاولة ونشر وتطوير محتوى عربي في كافة المجالات , سأقوم بإذن الله بشرح كتاب "مقدمة في الخوارزميات" الإنجليزي بلغة بسيطة مفهومة وسأضع خطة لإنهاء هذا العمل في خلال سنتين من الآن والله المستعان



## الفصل الثالث

أكتوبر 19, 2018

### Growth of functions

#### نمو الدوال

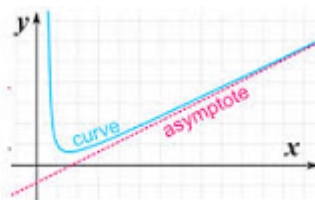
#### مقدمة

في الفصل الثاني اتعرضنا لبعض الalgorithms التي يتحل مشكلة الsorting واتكلمنا عن الrunning time لكل algorithm ولكن أجلبنا الكلام عن ازاى نعبر عن الrunning time باستخدام الnotations , في الفصل ده هنتعلم ازاى نعبر عن الrunning time لاي algorithm بشكل precise (دقيق جدا).

نفهم عنوان الفصل اللي بنتكلم فيه , **functions** يعني دوال وده لأنه بيستخدم بعض الدوال للتعبير عن الrunning time للalgorithms , بعد تعريف الدوال دي بيستخدمها ك **notation** بعد كدة على مدار الكتاب , وكلمة **growth** يعني نمو أو زيادة الدخل , وده لإن أهم حاجة وانت بتمثل الrunning time ان التمثيل ده يكون بيوصف الrunning time صح مهما الinput يزيد , يعني من الآخر هنستخدم بعض الدوال للتعبير عن كفاءة الalgorithm او الrunning time للalgorithm عند زيادة الدخل , وليه عند زيادة الدخل؟ ده لإن أي algorithm ممكن يكون أداءه سريع على input صغير ولكن الalgorithm بتبان مدى كفاءته عند زيادة الدخل لأرقام كبيرة جدا. اتعرضنا في الفصل الثاني لمفهوم ال **order of growth** , وكلمة **order** لما تيجي وسط الكلام عن الرياضيات والمعادلات بتعني أس , ال **order of growth** هو الأس اللي له أكبر تأثير على الخرج عند زيادة الدخل , وهو طبعا أكبر أس في المعادلة , يعني مثلا لو عندنا المعادلة دي  $3n^3$  ال **order of growth** هو 3 فهو ده ال **order of growth** , طبعا واضح إن أكبر أس في المعادلة هو 3 فهو ده ال **order of growth** , هو اللي بيأثر عند زيادة الدخل بمعنى ان لو الدخل وصل 1000 فقيمة باقي المعادلة (5000010) بتساوي قليل جدا بالنسبة لقيمة ال  $n^3$  أعلى أس (3000000000) يعني قيمة باقي المعادلة تمثل 0.0017 من قيمة ال **order of growth** لدرجة يمكن إهمال باقي المعادلة دي عند تمثيل الrunning time.

### Asymptotic notation

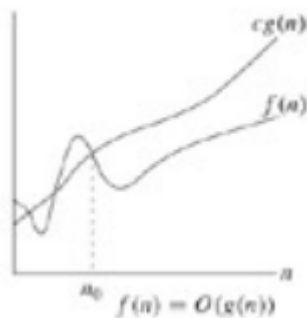
كلمة **asymptotic** جاية من كلمة **asymptote** ومعناها **line** (خط مستقيم) يقترب جدا من **curve** (منحنى) بس عند زيادة الدخل , أما عند الدخل الصغير يكون بعيد عنه زي مانت شايف في الصورة.



يعني ايه بقى **asymptotic notation**؟ يعني notation يقترب جدا من القيمة الحقيقية لل **running time** عند زيادة الدخل , ولكن عند دخل صغير يكون بعيد. وبنفس المعنى **asymptotic efficiency** يعني الكفاءة عند زيادة الدخل , وال **asymptotic analysis** التحليل عند زيادة الدخل , بيقولك معلومة ان ال **algorithm** الاكثر كفاءة **asymptotically** هو أفضل اختيار إلا مع الدخل الصغير, هنعرف باذن الله **5 Asymptotic notations** :

## 1) Big O notation

ال **O notation** هي function بتستخدم لتحديد ال **upper bound** (الحد الأقصى) لأي function ثانية , والحد الأقصى مش بالضرورة يكون حاجة واحدة بس بمعنى ان انا ممكن احدد اكثر من حد لدالة معينة متقدرش تعدي ولا واحد فيهم , يعني كمثال  $f(n) = 5n^2 + 11n + 13$  ال **O notation** ليها  $O(n^2)$  وده رياضيا معناه ان فيه **constant** لو ضربته في  $n^2$  الدالة  $f(n)$  متقدرش تعديه لما يكون الدخل أكبر من قيمة معينة بنسميها  $n_0$  , والصورة اللي جاية بتمثل ال **O notation** .

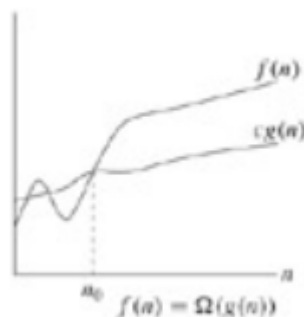


لما نكتب  $f(n) = O(n^2)$  أو أي notation ثاني احنا بنستخدم ال = بمعنى ينتمي إلى يعني الدالة  $f(n)$  بتنتمي للمجموعة (set) المتمثلة بـ  $O(n^2)$  أو أي notation زي ماقلنا. الحد اللي بتعرفه دالة ال  $O$  فيه منه نوعين : نوع بنسميه **asymptotically tight** وده معناه ان الحد ده ممكن الدالة تساويه زي المثال اللي فات  $f(n) = 5n^2 + 11n + 13 = O(n^2)$  الدالة فيها أس  $n^2$  والحد الأقصى ليها  $n^2$  وبالتالي في علاقة تساوي بينهم ممكنة  $f(n) \leq O(n^2)$  , أما النوع الثاني هو عكسه **not asymptotically tight** وده معناه ان الدالة لا يمكن تساوي أو تلامس الحد , كمثال  $f(n) = 5n^2 + 11n + 13 = O(n^3)$  ال  $n^3$  برده حد أقصى لا يمكن الدالة  $f(n)$  تعديه ولكنها لا يمكن توصله برده وبالتالي العلاقة بينهم مفهوش تساوي  $f(n) < O(n^3)$  .

ال **O notation** بنستخدمه في إيه في ال algorithms بنستخدمه في التعبير عن ال **Worst-case running time** لأن احنا في ال worst-case بنحتاج نحدد أقصى (upper) وقت ممكن ياخذ ال algorithm في حل المشكلة , كمثال نقدر نعبر عن ال worst-case running time ل insertion sort ب  $T(n) = O(n^2)$  زي ماعرفنا فالفصل الثاني.

## 2) Big Omega notation

ال **Ω notation** هي function بنستخدم لتحديد ال **lower bound** (الحد الأدنى) لأي function ثانية , ونفس الكلام فيه أكثر من حد للدالة , نفس المثال  $f(n) = 5n^2 + 11n + 13$  ال notation ليها  $f(n) = Ω(n)$  , ونفس الطريقة ده رياضيا معناه ان في constant لو ضربته في  $n$  الدالة  $f(n)$  متقدرش تعديه لما يكون الدخل أكبر من قيمة معينة بنسميها  $n_0$  , والصورة اللي جاية بتمثل ال **Ω notation**

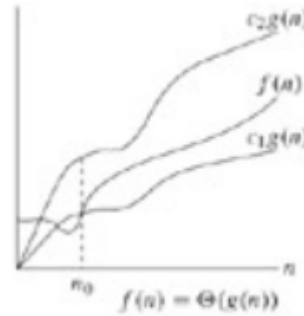


وبنفس الطريقة فيه نوعين لل **bound** هما **asymptotically tight** والثاني **not asymptotically tight** وتم شرحهم ولكن الفرق ان ال **Ω notation** بتحدد الحد الأدنى فبالنالي العلاقة  $f(n) > Ω(n)$  او  $f(n) ≥ Ω(n)$ .

ال **Ω notation** بنستخدم في إيه في ال algorithms بنستخدم في التعبير عن ال **best-case running time** لأن احنا في ال best-case بنحتاج نحدد أقل (lower) وقت ممكن ياخذ ال algorithm لحل المشكلة , كمثال نقدر نعبر عن ال best-case running time ل insertion sort ب  $T(n) = Ω(n)$  زي ماعرفنا فالفصل الثاني.

## 3) Theta notation

ال **Θ notation** هي عبارة عن مجموع ال **O notation** وال **Ω notation** يعني هي دالة بتحدد **lower and upper bounds** ل function ثانية , كمثال  $f(n) = 15n + 3$  ال  $Θ$  notation هيكون  $g(n) = Θ(n)$  , وفي الحالة دي بنسمي الدالة  $g(n)$  انها **asymptotically tight bound** للدالة  $f(n)$  , وده معناه رياضيا ان فيه 2 constants لو ضربتهم في  $n$  يمثلو **upper and lower bounds** للدالة  $f(n)$  لما يكون الدخل أكبر من قيمة معينة بنسميها  $n_0$  , والصورة دي بتوضح ال **Θ notation**



لو قلت ان  $f(n) = \Theta(n)$  فتقدر تستنتج حاجتين

$$(f(n) = O(n), f(n) = \Omega(n))$$

والعكس صحيح يعني لو اثبتت ال  $O$  وال  $\Omega$  وطلعو نفس ال function تقدر تستنتج ال  $\Theta$ .

#### 4) Little o notation

ال **o notation** هي بالظبط ال  $O$  notation بس ال bound بتاعها بيكون not asymptotically tight بس ومينفعش يكون asymptotically tight , كمثال

$$f(n) = 19n^3 + 4n^2 + 8n + 3$$

$$f(n) = O(n^4), f(n) = O(n^3)$$

$$f(n) = o(n^4), f(n) \neq o(n^3)$$

يعني ال **o notation** بيحدد **upper bound** للدالة لكن الدالة لا يمكن توصل لل bound ده (not asymptotically tight)

#### 5) Little ω notation

ال **ω notation** بالنسبة لل  $\Omega$  notation بالظبط زي مال  $o$  notation بالنسبة لل  $O$  notation , يعني هي بتحدد **lower bound** برده بس not asymptotically tight , كمثال

$$f(n) = 10n^2$$

$$f(n) = \Omega(n), f(n) = \Omega(n^2)$$

$$f(n) = \omega(n), f(n) \neq \omega(n^2)$$



أدخل تعليقك...



المشاركات الشائعة من هذه المدونة

## الفصل الخامس

أكتوبر 19, 2018

### Probabilistic Analysis and Randomized Algorithms

مقدمة في الفصل الرابع اتعرضنا لما يسمى الـ **recurrence** وقلنا انه عبارة عن معادلة بتوصف **function** بدلالة قيمتها مع دخل أصغر أو بمعنى ثاني بتوصف **recursive function** يعني دالة بتنادي نفسها مع دخل أصغر لغاية متوصل لدخل صغير جدا تقدر تحل المشكلنا ...

```

ANT(n)
    → candidate
    1 to n
    interview candid
    if candidate i i
    then best = i
    hire can
  
```

[قراءة المزيد](#)

## الفصل الأول

أكتوبر 19, 2018

### The Role of Algorithms in Computing دور الـ **algorithms** في علم الكمبيوتر

**Algorithms** يعني إيه **algorithm**? الـ **algorithm** هو ببساطة طريقة لحل مشكلة معينة , أو بشكل أدق هو مجموعة من الـ **steps** (خطوات) لحل مشكلة. يعني إيه **problem** (مشكلة)? مشكلة يعني عندي **input** (دخل) عايز احوله لـ **output** (خرج) وفق **relationship** (علاقة) واضحة بينهم. مثال : مشكل ...

[قراءة المزيد](#)

## الجزء الأول

أكتوبر 19, 2018

## Foundations الأساسيات

الجزء الأول من الكتاب يتكلم عن الأساسيات (يعني إيه algorithms , أهمية ال algorithms وهكذا) كمان بيبيّن الأساس اللي هيمشي عليه في الكتاب . الفكرة الأساسية للجزء هي **designing and analyzing algorithms** , أولا إيه الفرق بين ال **design** وال **analysis** ? ببساطة ال **design** ان عندك مشكلة وا ...

[قراءة المزيد](#) يتم التشغيل بواسطة Blogger

صور المظاهر بواسطة Michael Elkan



HOSSAM ABDELNASER

[الانتقال إلى الملف الشخصي](#)



الإبلاغ عن إساءة الاستخدام