Learning Algorithms From Scratch / Algorithms for Dummies

Dec 22nd, 2011 12:00 am | Comments

When you are programming you face challenges all the way. Getting the problems solved is just the tip of the iceberg, getting it done efficiently is the rest.

Update: graphs are gone in this post, sorry. I re-made this post and added more information and images. Checkout it out at: http://adrianmejia.com/blog/2014/02/13/algorithms-for-dummies-part-1-sorting/

Why should you care for efficiency?

Solutions to the same problem might take years with certain algorithm, and just minutes using efficient algorithms. For instance, if you have applications that are used for thousands of people over internet, every fraction of second counts. Therefore, efficient algorithms is a must.

How I do my algorithms more efficient?

To improve something you first need to know the actual state. In this case you need to measure the actual effectiveness of your algorithm in other to improve it. It's very common to use running time analysis to measure the speed of algorithms independently from the hardware used (old pc, supercomputer it doesn't matter).

Run-time analysis

A common way to analyze the algorithms is using the big-O notation. The good thing about this notation is that is independent from the computer used to run the algorithm. You know that if you use a very slow computer (e.g. pentium I) v.s. a supercomputer use in NASA, the latter will run the program much faster. Big-O notation abstract the hardware and just focus in the algorithm per se. The only variable in the big-O notation gives the relative time needed to process an algorithm in function of the input n. Let's clarify this with an example.

Ex.1 - You want to sort an array A of n integers.

Depending in the algorithm used to do that you may have:

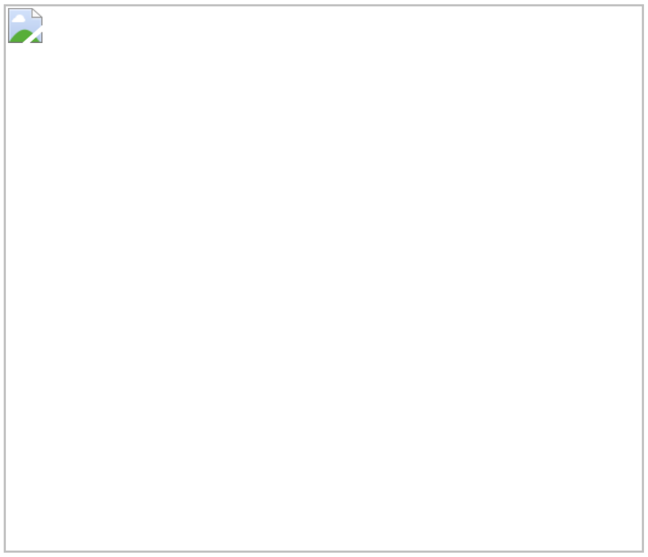
- **selection** sort has a running time of O(n^2);
- **merge sort** –> O(n log n)

Right now, it doesn't matter if are not familiar with these algorithms (we will cover this the next lessons), the point here is that we have n integer and big-O notations give us a mathematical expression that is in function of the input n. If you plot in a graph n^2 and n log n. You'll see that n^2 grows much faster than n log(n). That means that the algorithm n^2 will take longer than n*log(n) to process as the size of the array n increases.

Common order of Growth

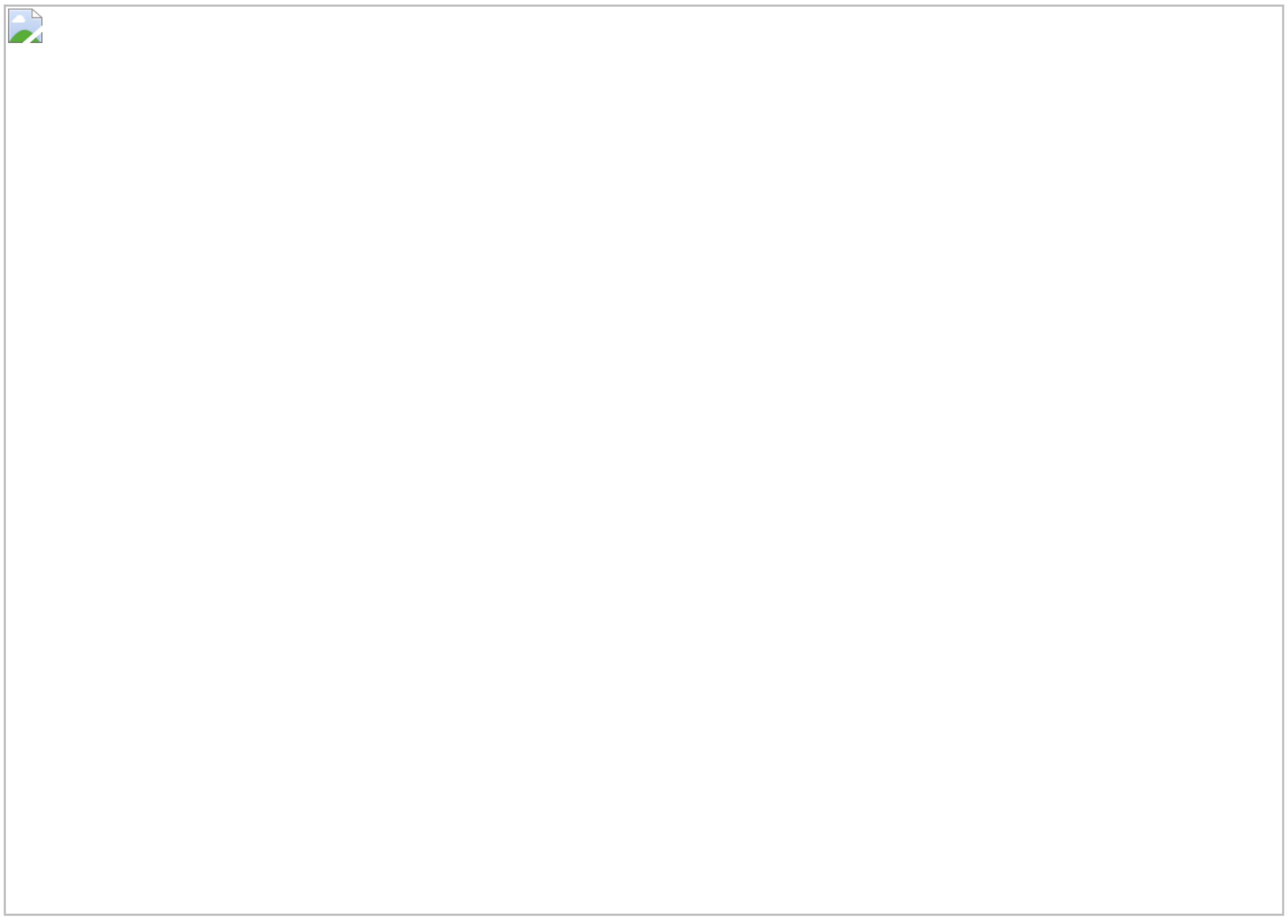
To give you an idea of the common order of growth of runtime expressions. Take a look at the following graph and table. The slower the function growth the better is the algorithm. In order from better performance to worst is:

1 – log n – n – n log n – n^2 – n^3 – 2^n – n! …



Approximate growth rate from code.

There are a whole theory and math behind the Big-O notation and other notations related. At this time, just take a look of the typical code and its growth order.



Cases (the good, the bad, and the ugly)

Remember that n is the number of elements in the input. All this runtime growth rate are in function of the input elements. There is another important thing to consider about the input elements: the order! The order of the input elements matters, and that's why algorithms are analyzed in 3 different cases:

1. Worst-case performance: the input is distributed as worst as it could be for an algorithm.
2. Average-case scenario: approximation of the most common arrange of inputs.
3. Best-case scenario: most favorable distribution of the inputs.
4. One more: Space. this is how much space the algorithm cosume to execute.

If you want more depth in these topic read here:

- Analysis (pdf) (keynote)
- Algorithm @ ocw.mit.edu: lectures 1 and 2
- http://algs4.cs.princeton.edu/home/

Posted by Adrian Mejia Dec 22nd, 2011 12:00 am algorithms, analysis of algorithm, big-o, for dummies, runtime analysis

Tweet G+1 2 Like Share Be the first of your friends to like this.

« How to remove programs from the start up in Mac OS X Concentration problems? Procastination? You're not the only one. »

Comments

7 Comments Adrian Mejia's Blog 1 Login

Recommend Share Sort by Best

Join the discussion…

Rafael · 2 years ago

Great info, but all graphs are gone

3 · Reply · Share ›

Learner · 3 years ago

Great read, teaches me something new. Thanks and please keep up the good work.

1 · Reply · Share ›

Richard Johnson · 2 years ago

my name is Richard I'm a visual learner if anybody is willing to teach me basic algorithms I'm willing to learn my number is 214 866-9654 please call me

• Reply • Share ›

Zing • 3 years ago
Informative, but your grammar needs tweaked.
• Reply • Share ›

Nathan Nash ➦ Zing • 3 years ago
Forgot the '-ing' there buddy.
5 • Reply • Share ›

Student • 3 years ago
Thank you very much
• Reply • Share ›

Inexplicable_apathy • 3 years ago
Thanks for this, it was a good and easy to understand read.
• Reply • Share ›

ALSO ON **ADRIAN MEJIA'S BLOG** WHAT'S THIS?

Grunt JS Tutorial From Beginner to Ninja
5 comments • a year ago
Shuklendu Ayachit — While executing currency example. I am getting below error. Could you please tell me cause and solution to

Algorithms for Dummies (Part 1): Big-O Notation and Sorting
9 comments • 2 years ago
Odumba500 — is there anything on reverse order of a sentence and how to write step by step instructions? without code?

Cheap Airplay Receiver With Raspberry Pi - Adrian Mejia's [code]Blog
6 comments • 2 years ago
Scott — A bit of research and I got it working by doing the following. (Now my AirPi starts automatically on boot) STEP 1. Edit

Microsoft Zune Failure Analysis
1 comment • 2 years ago
tomcarrdpi — Got this 30GB Zune for a steal, especially for the storage size - http://www.amazon.com/dp/B000F... it has

Subscribe Add Disqus to your site Add Disqus Add Privacy

Recent Posts

- How to Build Scalable Apps in 2016?
- Grunt JS Tutorial From Beginner to Ninja
- MEAN Stack Tutorial MongoDB ExpressJS AngularJS NodeJS (Part III)
- Creating RESTful APIs With NodeJS and MongoDB Tutorial (Part II)
- AngularJS Tutorial for Beginners With NodeJS ExpressJS and MongoDB (Part I)

RSS

adrianmejia

@amejiarosario

@amejiarosario

Google+

Subscribe

Enter your email to receive updates

you@email.com

Subscribe

GitHub Repos

- meanshop

  Building an e-commerce application with the MEAN stack

- amejiarosario.github.io

- impressjs-presentations

- deprecated-meanshop

  Building an e-commerce application with the MEAN stack

- blogx

@amejiarosario on GitHub