ANALYSIS OF ALGORITHMS

LECTURE 2 : ASYMPTOTIC ANALYSIS

ASYMPTOTIC ANALYSIS

- Before we start, pause the video and read Section 2.3 in the book

ASYMPTOTIC ANALYSIS

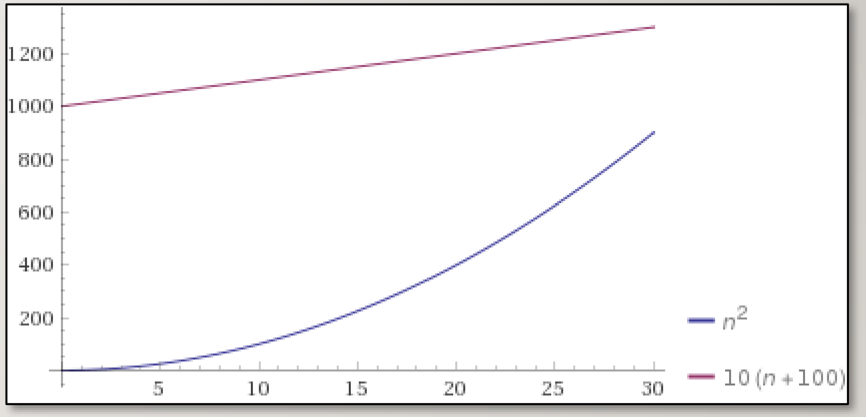
- I’ll wait…

ASYMPTOTIC ANALYSIS

- Anything seem a bit weird to you?
- Yeah, in example 2.3 it says that line 4 is executed 11 times. Why?
    - How many times does the loop execute?
    - But we need one more to determine that we should leave the loop
- Any other questions about section 2.3?
- Note that we're trying to find an equation relating the problem size *n* to the time taken *t*
- Note that I'm not covering it in a lecture not because it's not important (It most definitely is) or because it's not in any tests (It most certainly is), but more because it doesn't need more explanation than the book provides

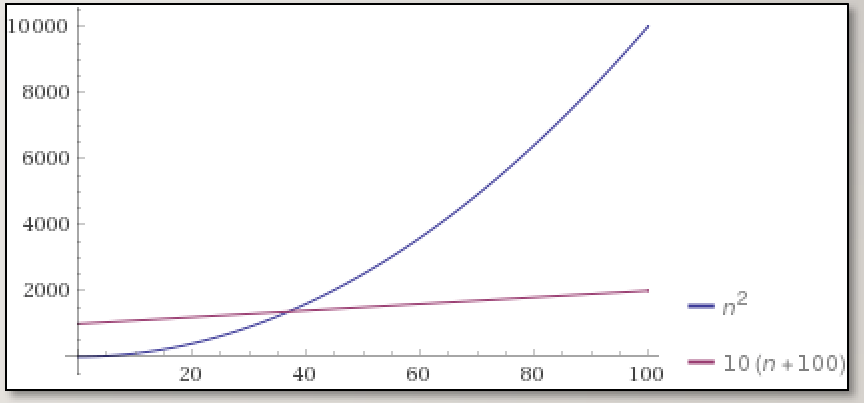
ASYMPTOTIC ANALYSIS

- What is asymptotic analysis?
  - Trying to find an equation describing the relationship between problem size and time taken, but we only really care about large values of n
- Why do we do asymptotic analysis?
- Some geometric intuition
- Which is worse, $10n + 1000$ or n^2?
  - Note that greater is worse, because we're talking about time taken



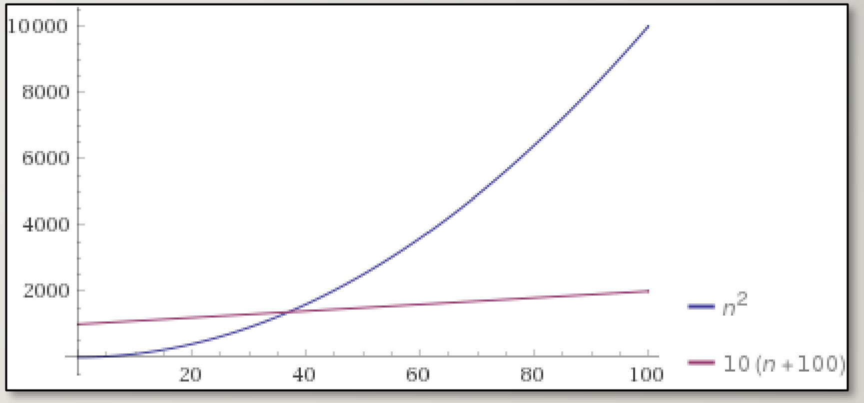
ASYMPTOTIC ANALYSIS

- What is asymptotic analysis?
- Why do we do asymptotic analysis?
- Some geometric intuition
- Which is worse, $10n + 1000$ or n^2?
  - Asymptotic analysis only cares about which is eventually bigger



ASYMPTOTIC ANALYSIS

- What is asymptotic analysis?
- Why do we do asymptotic analysis?
- Some geometric intuition
- Which is worse, $10n + 1000$ or n^2?
  - Asymptotic analysis only cares about which is eventually bigger
- Is it valid to do that?
  - Small problems can be solved by anything so who cares? We only really get involved when the problem is big and scary



ASYMPTOTIC ANALYSIS

- Complexity classes exist to define the asymptotic bounds of the function describing the performance. Note that these are sets of functions.
- $O(f(n))$
  - Upper bound – all function in this class are no worse than $f(n)$
- $\Omega\big(f(n)\big)$
  - Lower bound – all function in this class are no better than $f(n)$
- $\Theta\big(f(n)\big)$
  - Exact bound – all functions in this class are no better and no worse than $f(n)$

ASYMPTOTIC ANALYSIS – UPPER BOUNDS

- $g(n) \in O(f(n))$
  - What we're trying to express here is that $g(n)$ isn't larger than $f(n)$ for large n
  - So we want to express that we only care about large n, and we want to express that we don't really care about constant factors
- $g(n) \in O\big(f(n)\big) \Leftrightarrow \exists c, n_0 \ni g(n) \leq cf(n) \; \forall n \geq n_0$
  - Um.. What?
  - We're defining a condition for membership of the set $O(f(n))$.
  - The condition is that there is some crossing point n_0 after which $g(n)$ is not larger than $f(n)$.
  - The *c* term allows us to ignore constant factors

ASYMPTOTIC ANALYSIS – LOWER BOUNDS

- $g(n) \in \Omega(f(n))$
  - $g(n)$ isn't smaller than $f(n)$ for large n
- $g(n) \in \Omega\big(f(n)\big) \Leftrightarrow \exists c, n_0 \ni g(n) \geq cf(n)\ \forall n \geq n_0$
  - Wait this isn't so bad…
  - We're defining a condition for membership of the set $\Omega(f(n))$.
  - The condition is that there is some crossing point n_0 after which $g(n)$ is not smaller than $f(n)$.
  - Again, the *c* term allows us to ignore constant factors

ASYMPTOTIC ANALYSIS – COMBINED

- $g(n) \in \Theta(f(n))$
  - $g(n)$ isn't smaller or bigger than $f(n)$ for large n
- $g(n) \in \Omega\big(f(n)\big) \Leftrightarrow g(n) \in (O\big(f(n)\big) \cap \Omega\big(f(n)\big))$
  - g(n) is in both $O(f(n))$ and $\Omega(f(n))$

SOME THOUGHTS

- Is $n^2 \in O(n)$?
- Is $n \in O(n^2)$?
- Is $n^2 \in \Omega(n)$?
- Is $n^2 + 2n + 3 \in O(n^2)$?
- Is $n^2 + 2n + 3 \in \Theta(n^2)$?
- Is $n \in \Theta(n^2)$?

SOME THOUGHTS

- If the best case complexity class $\in O(n^2)$, is it possible for some cases of the algorithm to be $\in \Omega(n^3)$?
- If the worst case complexity class $\in O(n^2)$, is it possible for some cases of the algorithm to be $\in \Omega(n^3)$?
- If the best case complexity class $\in O(n^2)$, is it possible for some cases of the algorithm to be $\in O(n^3)$?