Introduction to big-Oh

October 9, 2017

Definition: A theoretical measure of the execution of an algorithm, usually the time or memory needed, given the problem size n, which is usually the number of items. Informally, saying some equation f(n) = O(g(n)) means it is less than some constant multiple of g(n). The notation is read, ”f of n is big oh of g of n”.

Formal Definition: $f(n) = O(g(n))$ means there are positive constants $candk, suchthat 0 f(n) cg(n) forallnk$. The values of c and k must be fixed for the function f and must not depend on n.

1 Aim: Classification of Functions

- In computer science, we often need a way to group together functions by their scaling behavious, and the classification should
  - Remove unnecessary details
  - Be (rekatively) quick and easy
  - Be able to deal with ‘weir‘ functions that can happen for runtimes
  - Still be mathematically well-defined
- Experience of CS is that this is best done by the ‘big-Oh notation and family‘

2 Experimental Studies

General patern:

- Write a program implementing the algorithm
- Run the program with inputs of ”varying size and composition”
- Use a system method to get an (in)accurate measure of the actual running time
- Plot the results
- Intterpret & analyse