I00041

Information Retrieval

Block A

Big Oh notation

(based on material from Dan Melamed's course)

Big Oh notation / Big O notation

- Goal: evaluate the run-time behaviour of your algorithm independent of particular hardware
- Abstract away minor differences
- Compare the efficiency of programs in relation to the length of the input.

Example

- Finding a word in a list of words (dictionary) of size N
- If dictionary in unsorted
  - This will take N comparisons (each with a time *c1)*
- If dictionary is sorted
  - This will take Log2 (N) comparisons at max (each with time *c2)*
- Which algorithm is faster?
- Constants are unknown and largely irrelevant for big N

Big O

- The most common method and notation for discussing the execution time of algorithms is "Big O”.
- For the alphabetized dictionary the algorithm requires O(log N) steps.
- For the unsorted list the algorithm requires O(N) steps.
- Big O is the *asymptotic execution time* of the algorithm.

Big O Examples

- $3n^3 \Rightarrow O(n^3)$
- $3n^3 + 8 \Rightarrow O(n^3)$
- $8n^2 + 10n * \log(n) + 100n + 10^{20} \Rightarrow O(n^2)$
- $3\log(n) + 2n^{1/2} \Rightarrow O(n^{1/2})$
- $2^{100} \Rightarrow O(1)$
- $T_{linearSearch}(n) \Rightarrow O(n)$
- $T_{binarySearch}(n) \Rightarrow O(\log(n))$

- $T_{mergingpostinglists}(m,n) \Rightarrow O(m+n)$

Summing Execution Times

- If an algorithm's execution time is $N^2 + N$ then it is said to have $O(N^2)$ execution time, not $O(N^2 + N)$.

- When adding algorithmic complexities the larger value dominates.

- Formally, a function f(N) dominates a function g(N) if there exists a constant value n_0 such that for all values $N > N_0$ it is the case that $g(N) < f(N)$.

Ranking of Algorithmic Behaviors

Function	Common Name
$N!$	factorial
2^N	Exponential
N^d, $d > 3$	Polynomial
N^3	Cubic
N^2	Quadratic
$N\sqrt{N}$	
$N \log N$	
N	Linear
$\sqrt{N}$	Root - n
$\log N$	Logarithmic
1	Constant

Running Times

- Assume N = 100,000 and processor speed is 1,000,000 operations per second

Function	Running Time
2^N	over 100 years
N^3	31.7 years
N^2	2.8 hours
$N\sqrt{N}$	31.6 seconds
N log N	1.2 seconds
N	0.1 seconds
$\sqrt{N}$	3.2×10^{-4} seconds
log N	1.2×10^{-5} seconds

Grap