NPTEL MOOC

PROGRAMMING, DATA STRUCTURES AND ALGORITHMS IN PYTHON

Week 3, Lecture 5

Madhavan Mukund, Chennai Mathematical Institute
http://www.cmi.ac.in/~madhavan

Efficiency

- Measure time taken by an algorithm as a function T(n) with respect to input size n
- Usually report worst case behaviour
  - Worst case for searching in a sequence is when value is not found
  - Worst case is easier to calculate than “average” case or other more reasonable measures

O( ) notation

- Interested in broad relationship between input size and running time
- Is T(n) proportional to log n, n, n log n, n^2, …, 2^n?
- Write T(n) = O(n), T(n) = O(n log n), … to indicate this
  - Linear scan is O(n) for arrays and lists
  - Binary search is O(log n) for sorted arrays

Typical functions T(n)…

Input	log n	n	n log n	n^2	n^3	2^n	n!
10	3.3	10	33	100	1000	1000	10^6
100	6.6	100	66	10^4	10^6	10^{30}	10^{157}
1000	10	1000	10^4	10^6	10^9		
10^4	13	10^4	10^5	10^8	10^{12}		
10^5	17	10^5	10^6	10^{10}			
10^6	20	10^6	10^7				
10^7	23	10^7	10^8				
10^8	27	10^8	10^9				
10^9	30	10^9	10^{10}				
10^{10}	33	10^{10}					

Python can do about 10^7 steps in a second

Efficiency

- Theoretically $T(n) = O(n^k)$ is considered efficient
  - Polynomial time
- In practice even $T(n) = O(n^2)$ has very limited effective range
  - Inputs larger than size 5000 take very long