

Estimated Time to Sum Subarrays

- · Ignore data initialisation
- "Brute-force" summing with two nested loops:

$$T(n) = m(m+1) = {n \choose 2} ({n \choose 2} + 1)$$

= 0.25 n^2 + 0.5 n

- For a large n, $T(n) \approx 0.25n^2$
 - e.g., if $n \ge 10$, the linear term $0.5n \le 16.7\%$ of T(n)
 - if $n \ge 500$, the linear term $0.5n \le 0.4\%$ of T(n)

Lecture 2

COMPSCI 220 - AP G Gimel'farb



Quadratic vs linear term

$T(n) = 0.25n^2 + 0.5n$				
n	T(n)	$0.25n^2$	0.5n	
10	30	25	5	16.7%
50	650	625	25	3.8%
100	2550	2500	50	2.0%
500	62750	62500	250	0.4%
1000	250500	250000	500	0.2%

Lecture 2

COMPSCI 220 - AP G Gimel'farb





Quadratic Time to Sum Subarrays: $T(n)=0.25n^2+0.5n$

- Factor c = 0.25 is referred to as a "constant of proportionality"
- An actual value of the factor does not effect the behaviour of the algorithm for a large *n*:
 - 10% increase in $n \rightarrow 20\%$ increase in T(n)
 - Double value of $n \rightarrow 4$ -fold increase in T(n):

$$T(2n) = 4 T(n)$$

Lecture 2

COMPSCI 220 - AP G Gimel'farb



Running Time: Estimation Rules

- Running time is proportional to the **most** significant term in T(n)
- Once a problem size becomes large, the most significant term is that which has the largest power of n
- This term increases faster than other terms which reduce in significance

Lecture 2

COMPSCI 220 - AP G Gimel'farb





Running Time: Estimation Rules

- Constants of proportionality depend on the compiler, language, computer, etc.
 - It is useful to ignore the constants when analysing algorithms.
- Constants of proportionality are reduced by using faster hardware or minimising time spent on the "inner loop"
 - But this would not effect behaviour of an algorithm for a large problem!

Lecture 2

COMPSCI 220 - AP G Gimel'farb





Elementary Operations

- Basic arithmetic operations (+ ; ; * ; / ; %)
- Basic relational operators (==, !=, >, <, >=, <=)
- Basic Boolean operations (AND,OR,NOT)
- Branch operations, return, ...

Input for problem domains (meaning of *n*):

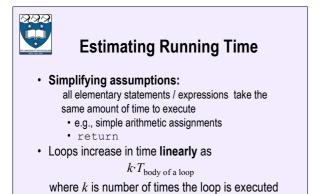
Sorting: n items Graph / path: n vertices / edges Image processing: n pixels Text processing: string length

Lecture 2

COMPSCI 220 - AP G Gimel'farb



Lecture 2



COMPSCI 220 - AP G Gimel'farb

