



Design and Analysis of Algorithms

Lecture - 2

Success is always inevitable with Hard Work and Perseverance

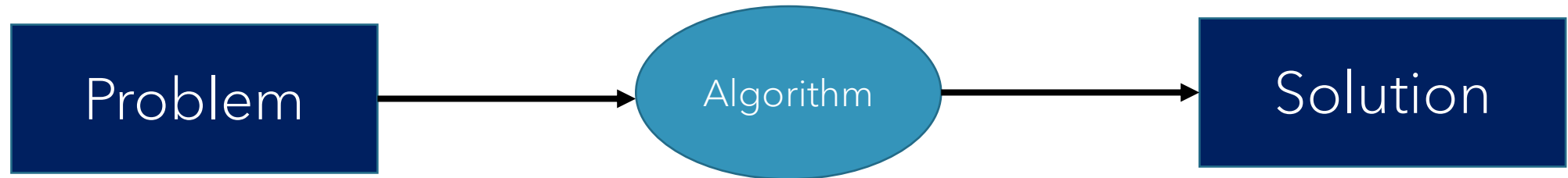
N. Ravitha Rajalakshmi

Learning Objective

- Learn mechanisms to effectively debug the programs
- Understand the framework used for measuring the runtime of the algorithm.

Problem Solving

How do we devise an algorithm ? (or) What is the primary criteria for an algorithm?



Correctness is important
Solution should be universal
For any problem instance, it should yield correct solution

Debugging

- Finding logical errors are always difficult
- Errors happen as the user is concerned with known test cases
- **Corner cases (or) edge cases** [input cases that lie at the extreme of the problem space] are not looked upon

Stress Test

- Random generation of test cases
- Comparing the output of the algorithm with some well known method

Maximum Pairwise sum

- Find the maximum sum of two distinct numbers in a sequence of non-negative integers.

Input: A sequence of non-negative integers.

Output: The maximum value that can be obtained by multiplying two different elements from the sequence.

Input: [9, 3, 10, 11, 8, 6]

Output: 21

Maximum Pairwise sum

		j				
	9	3	10	11	8	6
i	9		12	19	20	17
	3			13	14	11
	10				21	18
	11					19
	8					
	6					

Function MaxPairwiseSum (a, n)

a is array # n - number of elements

best = -1

for i from 1 to n-1

for j from i+1 to n

if (a[i] + a[j]) > best

best = a[i] + a[j]

return best

Pause & Think

- Stress test helps in identifying logical (or) syntactic errors?
- How do you detect
 - Syntactic errors -
 - Runtime errors -

Computing Runtime of an algorithm

- Actual amount of time taken by the algorithm for producing the output.
- Why to compute runtime?
 - Find which algorithm to be better
- Can lines of code be a metric for measuring the runtime?
 - No, as each line performs varied number of operations
- Can we use profilers?

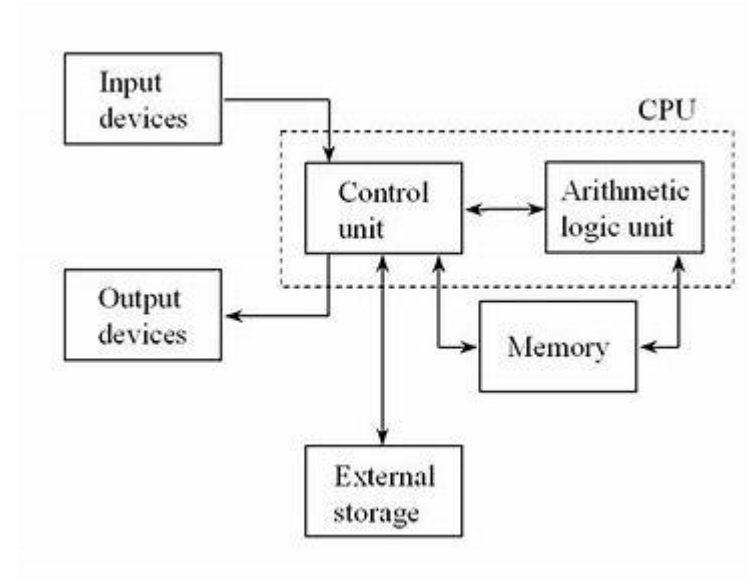
Why Measuring runtime is difficult??

- It depends on extraneous factors

Speed



System Architecture

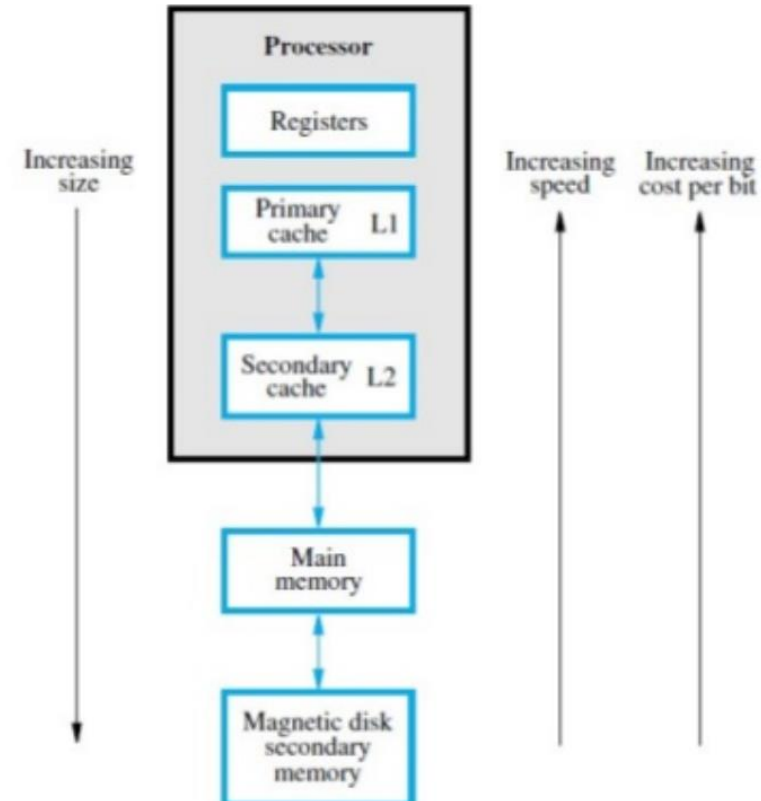


Why Measuring runtime is difficult??

Compiler Optimization



Memory Hierarchy



- Finding actual runtime is difficult
- Client specification is unknown

Measure runtime without considering these
unknowns

Pause & Think

- Is there anything else to be considered ?
- When will the algorithms differ ?

$5+7$

$5+7+89+56+12+1+56789+90+1+16$



Formal Framework to compute runtime

- Asymptotic Notation
- Measure of runtime that ignore the constant multiples
- Measure the runtime with respect to the growth of the input size

Approximate Runtime

Assume processing speed 1GHZ

It can perform 10^9 operations per second

n/Runtime	n	$n \log n$	n^2	2^n
$n = 20$	1sec	1sec	1sec	1sec
$n = 50$	1sec	1sec	1sec	13 <i>day</i>
$n = 10^2$	1sec	1sec	1sec	$4 \cdot 10^{13}$ year
$n = 10^6$	1sec	1sec	17 min	
$n = 10^9$	1sec	30sec	30 <i>year</i>	

Summary

- Logical errors to some extent can be easily identified through stress test
- Computing exact runtime is difficult
- Runtime is measured with respect to growth in input size

Thank You
Happy Learning

Success is always inevitable with Hard Work and Perseverance