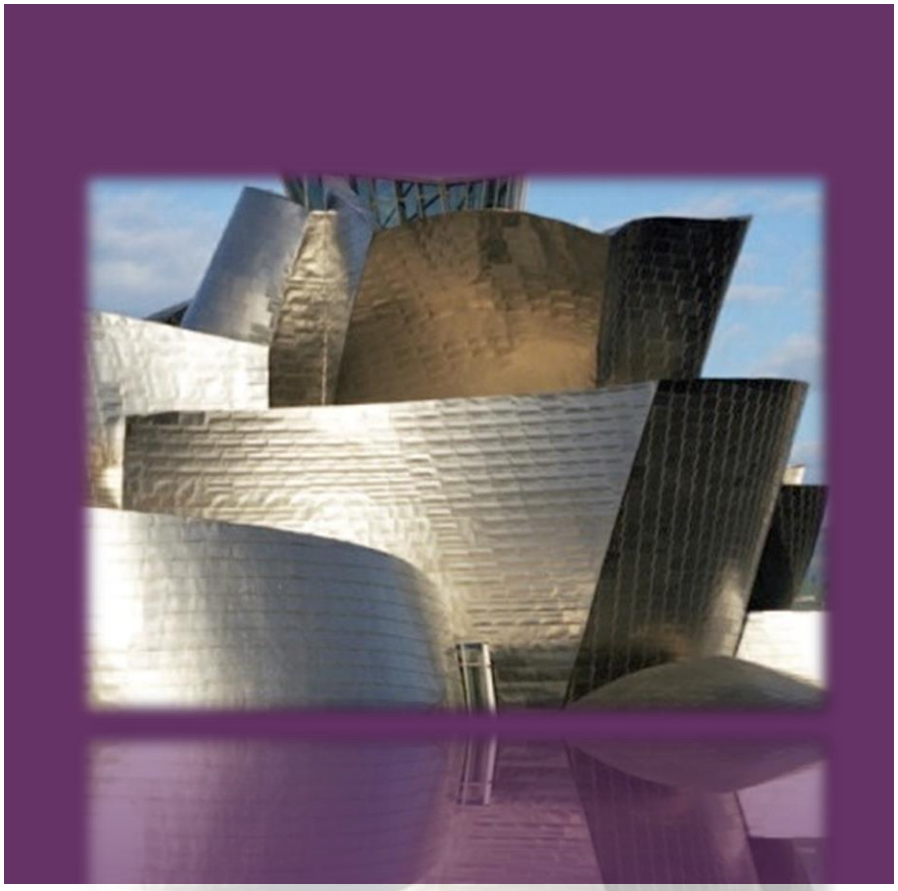


SECR2033
Computer Organization and Architecture

Lecture slides prepared by “Computer Organization and Architecture”, 9/e, by William Stallings, 2013.

Module 8
Performance Measurement and Analysis

Objectives:

- To understand the key performance issues that relate to computer design.
- Explain the reasons for the move to multicore organization, and understand the trade-off between cache and processor resources on a single chip.
- To summarize some of the issues in computer performance assessment.
- To discuss the benchmarks in general.

Module 8
Performance Measurement and Analysis

Module 8
Performance Measurement and Analysis

Overview

- Computer performance assessment is a quantitative science.
- Mathematical and statistical tools give us many ways in which to rate the overall performance of a system and the performance of its constituent components.
- There are so many ways to quantify system performance.

- In this section, we describe the most common measures of "average" computer performance and then provide situations where the use of each is appropriate and inappropriate.

- In comparing the performance of two systems, we measure the *time* that it takes for each system to perform the same amount of work.
- If the same program is run on two different systems, System A and System B:

  - System A is *n times as fast* as System B if:

$$\frac{Running\ time\ on\ system\ B}{Running\ time\ on\ system\ A} = n$$

  - System A is *x% faster* than System B if:

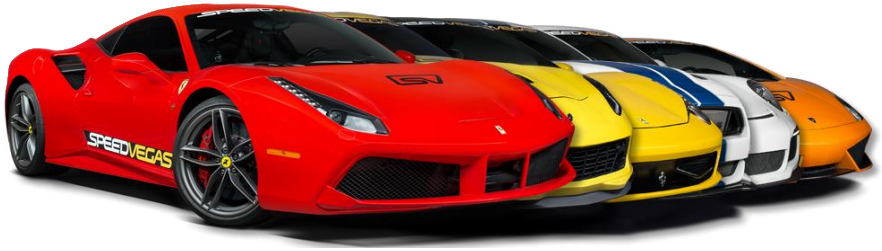
$$\left(\frac{Running\ time\ on\ system\ B}{Running\ time\ on\ system\ A} - 1\right) \times 100 = x$$

Example 7:

Consider the performance of two race cars. Car A completes a 10-mile run in 3 minutes, and car B completes the same 10-mile course in 4 minutes.

(a) How many time car A fast as car B?

(b) What is % car A faster than car B?



Solution :

Car A is 1.33 times as fast as Car B.

(a) How many time car A fast as car B?

$$\frac{Time\ for\ Car\ B\ to\ travel\ 10\ miles}{Time\ for\ Car\ A\ to\ travel\ 10\ miles} = \frac{4}{3} = 1.33$$

(b) What is % car A faster than car B?

$$= \left(\frac{Time\ for\ Car\ B\ to\ travel\ 10\ miles}{Time\ for\ Car\ A\ to\ travel\ 10\ miles} - 1 \right) \times 100$$
$$= (1.33 - 1) \times 100$$
$$= 0.33 \times 100$$
$$= 33\%$$

Car A is 33% faster than Car B.

8 Average Performance

- The previous formulas are useful in comparing the average performance of one system with the average performance of another.

- When we are evaluating system performance, we are most interested in its expected performance under a given workload.
- We use statistical tools, forming a concise *measures of central tendency*.

(a) The Arithmetic Mean

- The *arithmetic mean* (average) is the one with which everyone is most familiar.
- If we have n measurements, and we add them together and divide by n, then the result is the *arithmetic mean* (average), given by:

$$\frac{\sum_{i=1}^{n} x_i}{n}$$

- The arithmetic mean can be misleading if the data are skewed or scattered.

8

Example 8:

Consider the arithmetic average running time in seconds of five programs (v, w, x, y, z) on three systems (A, B, C) in the table below.

Program	System A Execution Time	System B Execution Time	System C Execution Time
v	50	100	500
w	200	400	600
x	250	500	500
y	400	800	800
z	5000	4100	3500
Average	1180	1180	1180

The performance differences are hidden by the simple average.

Example 9:

If execution frequencies (expected workloads) are known, a *weighted average* can be revealing. The running times for System A and System C are restated in the following table.

- Weighted average for System A = $(50 \times 0.5) + (200 \times 0.3) + (250 \times 0.1) + (400 \times 0.05) + (5000 \times 0.05) = 380$

Program	Execution Frequency	System A Execution Time	System C Execution Time
v	50%	50	500
w	30%	200	600
x	10%	250	500
y	5%	400	800
z	5%	5000	3500
Weighted Average		380 seconds	695 seconds

- Similar weighted average calculation for System C.
- System A is about 83% faster than System C for this particular workload.

Linda Null and Julia Lobur (2003). *The Essentials of Computer Organization and Architecture.* United States: Jones and Bartlett Publishers. p.454-455.

$$= \left(\frac{Weighted\ Average\ System\ C}{Weighted\ Average\ System\ A} - 1 \right) \times 100$$
$$= \left(\frac{695}{380} - 1 \right) \times 100$$
$$= (1.8289 - 1) \times 100$$
$$= 0.83 \times 100$$
$$= 83\%$$

Program	Execution Frequency	System A Execution Time	System C Execution Time
v	50%	50	500
w	30%	200	600
x	10%	250	500
y	5%	400	800
z	5%	5000	3500
Weighted Average		380 seconds	695 seconds

- System A is about 83% faster than System C for this particular workload.

- However, workloads can change over time.
- A system optimized for one workload may perform poorly when the workload changes, as illustrated below:

Example 10:

Program	System A Execution Time	Execution Frequency #1	Execution Frequency #2
v	50	50%	25%
w	200	30%	5%
x	250	10%	10%
y	400	5%	5%
z	5000	5%	55%
Weighted Average		380 seconds	2817.5 seconds

(b) The Geometric Mean

(Relative Performance)

- The previous discussion cannot use the arithmetic mean if the measurements exhibit a great deal of variability.
- The *geometric mean* gives us a consistent number with which to perform comparisons regardless of the distribution of the data.
- The geometric mean, G is defined as the nth root of the product, x_i of the n measurements, and represented by the following formula:

$$G = \sqrt[n]{x_1 \times x_2 \times x_3 \times \ldots \times x_n}$$

- Unlike the *arithmetic means*, the *geometric mean* does not give us a real expectation of system performance. It serves only as a tool for comparison.

- The geometric mean is more helpful than the arithmetic average when comparing the relative performance of two systems.

- The systems under evaluation are *normalized* to the reference machine when we take the ratio of the run time of a program on the reference machine to the run time of the same program on the system being evaluated.

Example 11: System A and System C normalized to System B.

Given the geometric means for a sample of five programs, found by taking the 5th root of the products of the normalized execution times for each system.

Program	System A Execution Time	Execution Time Normalized to B	System B Execution Time	Execution Time Normalized to B	System C Execution Time	Execution Time Normalized to B
v	50	2	100	1	500	0.2
w	200	2	400	1	600	0.6667
x	250	2	500	1	500	1
y	400	2	800	1	800	1
z	5000	0.82	4100	1	3500	1.1714
Geometric Mean		1.6733		1		0.6898

- The *geometric means* for System A and System C normalized to System B are calculated as follows:

$$
\begin{aligned}
G_A &= \sqrt[5]{\frac{100}{50} \times \frac{400}{200} \times \frac{500}{250} \times \frac{800}{400} \times \frac{4100}{5000}} \\
&= \sqrt[5]{2 \times 2 \times 2 \times 2 \times 0.82} \\
&= \sqrt[5]{13.12} \\
&= 1.6733
\end{aligned}
$$

$$
\begin{aligned}
G_C &= \sqrt[5]{\frac{100}{500} \times \frac{400}{600} \times \frac{500}{500} \times \frac{800}{800} \times \frac{4100}{3500}} \\
&= \sqrt[5]{0.2 \times 0.6667 \times 1 \times 1 \times 1.1714} \\
&= \sqrt[5]{0.1562} \\
&= 0.6898
\end{aligned}
$$

Example 12: System A and System B normalized to System C.

When another system is used for a reference machine, we get a different set of numbers.

Program	System A Execution Time	Execution Time Normalized to C	System B Execution Time	Execution Time Normalized to C	System C Execution Time	Execution Time Normalized to C
v	50	10	100	5	500	1
w	200	3	400	1.5	600	1
x	250	2	500	1	500	1
y	400	2	800	1	800	1
z	5000	0.7	4100	0.8537	3500	1
Geometric Mean		2.4258		1.4497		1

- The *geometric means* for System A and System B normalized to System C are calculated as follows:

$$
\begin{aligned}
G_A &= \sqrt[5]{\frac{500}{50} \times \frac{600}{200} \times \frac{500}{250} \times \frac{800}{400} \times \frac{3500}{5000}} \\
&= \sqrt[5]{10 \times 3 \times 2 \times 2 \times 0.7} \\
&= \sqrt[5]{84} \\
&= 2.4258
\end{aligned}
$$

$$
\begin{aligned}
G_B &= \sqrt[5]{\frac{500}{100} \times \frac{600}{400} \times \frac{500}{500} \times \frac{800}{800} \times \frac{3500}{4100}} \\
&= \sqrt[5]{5 \times 1.5 \times 1 \times 1 \times 0.8537} \\
&= \sqrt[5]{6.4028} \\
&= 1.4497
\end{aligned}
$$

... (Properties)

- One of the nice properties of the *geometric mean* is that we get the same results regardless of which system we pick for a reference.
- Based on previous 2 examples, we can notice that the ratio of the *geometric means* is consistent no matter which system we choose for the reference machine:

... (Properties)

Geometric mean tells us that "average improvement per program for system A when compared to system C is 2.43" NOT that "system A is 2.43 faster than system C"

System → *A* *B* *C*

Program	Execution Time Normalized to B	Execution Time Normalized to B	Execution Time Normalized to B
Geometric Mean	1.6733	1	0.6898

Program	Execution Time Normalized to C	Execution Time Normalized to C	Execution Time Normalized to C
Geometric Mean	2.4258	1.4497	1

Reference machine: *B*

$$\frac{Geometric\ mean\ A}{Geometric\ mean\ B} \approx \frac{Geometric\ mean\ A}{Geometric\ mean\ B}$$

$$\frac{1.6733}{1} \approx \frac{2.4258}{1.4497}$$

Reference machine: *C*

Reference machine: B

$$\frac{Geometric\ mean\ C}{Geometric\ mean\ B} \approx \frac{Geometric\ mean\ C}{Geometric\ mean\ B}$$

$$\frac{0.6898}{1} \approx \frac{1}{1.4497}$$

Reference machine: C

Reference machine: B

$$\frac{Geometric\ mean\ A}{Geometric\ mean\ C} \approx \frac{Geometric\ mean\ A}{Geometric\ mean\ C}$$

$$\frac{1.6733}{0.6898} \approx \frac{2.4258}{1}$$

Reference machine: C

Reference machine: B

$$\frac{Geometric\ mean\ B}{Geometric\ mean\ C} \approx \frac{Geometric\ mean\ B}{Geometric\ mean\ C}$$

$$\frac{1}{0.6898} \approx \frac{1.4497}{1}$$

Reference machine: C

Exercise 8.5:

Consider a sample of five programs execute on 3 different systems as shown in the table. If the System B and System C are normalized to System A:

(a) Fill in the execution time for each program in each system.

(b) Calculate the geometric means for all systems. Show your working.

Program	System A Execution Time	Execution Time Normalized to A	System B Execution Time	Execution Time Normalized to A	System C Execution Time	Execution Time Normalized to A
v	50		100		500	
w	200		400		600	
x	250		500		500	
y	400		800		800	
z	5000		4100		3500	

Solution :

$$G_B = \sqrt[5]{\frac{50}{100} \times \frac{200}{400} \times \frac{250}{500} \times \frac{400}{800} \times \frac{5000}{4100}}$$
$$= \sqrt[5]{0.5 \times 0.5 \times 0.5 \times 0.5 \times 1.2195}$$
$$= \sqrt[5]{0.0762}$$
$$= 0.5976$$

$$G_C = \sqrt[5]{\frac{50}{500} \times \frac{200}{600} \times \frac{250}{500} \times \frac{400}{800} \times \frac{5000}{3500}}$$
$$= \sqrt[5]{0.1 \times 0.3333 \times 0.5 \times 0.5 \times 1.4286}$$
$$= \sqrt[5]{0.0119}$$
$$= 0.4122$$

Program	System A Execution Time	Execution Time Normalized to A	System B Execution Time	Execution Time Normalized to A	System C Execution Time	Execution Time Normalized to A
v	50	1	100	0.5	500	0.1
w	200	1	400	0.5	600	0.3333
x	250	1	500	0.5	500	0.5
y	400	1	800	0.5	800	0.5
z	5000	1	4100	1.2195	3500	1.4286
Geometric Mean		1		0.5976		0.4122

Exercise 8.6:

Based on the previous example, If System A has been chosen as the reference machine, proof that the ratio of the geometric means are always consistent for System B and System C.

... (Problem with Geometric Mean)

- The inherent problem with using the geometric mean to demonstrate machine performance is that all execution times contribute equally to the result.
- So shortening the execution time of a small program by 10% has the same effect as shortening the execution time of a large program by 10%.
  - Shorter programs are generally easier to optimize, but in the real world, we want to shorten the execution time of longer programs.

- Also, the geometric mean is not proportionate → A system giving a geometric mean 50% smaller than another is not necessarily twice as fast!

Module 8
Performance Measurement and Analysis

8.1 Introduction
8.2 Defining Performance
8.3 Basic Measures of Cor
8.4 Comparing Performanc
8.5 Benchmarking
8.6 Summary

- Overview
- Clock Rate, MIPS and FLOPS
- Synthetic Benchmarks
- SPEC Benchmarks
- TPC Benchmarks
- System Simulation

8 Overview

- *Benchmarking* is the most common approach to assessing processor and computer system performance.
- *Performance benchmarking* is the science of making objective assessments of the performance of one system over another.
- Benchmarks are also useful for assessing performance improvements obtained by upgrading a computer or its components.

- Good benchmarks:
  - enable to cut through advertising hype and statistical tricks.
  - identify the systems that provide good performance at the most reasonable cost.

(a) Clock Rate, MIPS, and FLOPS

- CPU speed is a misleading metric that is most often used by computer vendors touting their systems' alleged superiority to all others.
- A widely metric related to clock rate is the $MIPS$.

Example 13:

- Saying that System A (1.4 GHz) is faster than System B (900 MHz) is valid only when the ISAs of System A and B are identical.
- With different ISAs, it is possible that both of these systems could obtain identical results within the same amount of wall clock time.

- There is a similar problem with the *FLOPS* metric.
- The FLOPS metric is even more difficult than the MIPS metric because there is no agreement as to what constitutes a floating-point operation.
- Despite their shortcomings, clock speed, MIPS, and FLOPS can be useful metrics in comparing relative performance across a line of similar computers offered by the same vendor.

8

(b) Synthetic Benchmarks

(*Whetstone*, *Linpack*, *Dhrystone*)

- The resulting execution time would lead to a single performance metric across all of the systems tested → *synthetic benchmarks*, because they don't necessarily represent any particular workload or application.

- Three of the better-known synthetic benchmarks are the *Whetstone*, *Linpack*, and *Dhrystone* metrics.

These programs are much too small to be useful in evaluating the performance of today's systems.

8

(c) SPEC Benchmarks

(*Standard Performance Evaluation Corporation*)

- The science of computer performance measurement benefited greatly by the contributions of the *Whetstone*, *Linpack*, and *Dhrystone* benchmarks.
- For one thing, these programs gave merit to the idea of having a common standard benchmark suite by which all systems could be compared → *SPEC benchmarks*.

- Their most widely known benchmark suite is the *CPU Suite*.
- The latest version of this benchmark is CPU2000 → consists of two parts, CINT2000, which measures integer arithmetic operations, and CFP2000, which measures floating-point processing.

SPEC Benchmark – CPU2017

from https://www.spec.org/cpu2017/results/cpu2017.html

All SPEC CPU2017 Results Published by SPEC

These results have been submitted to SPEC; see the disclaimer before studying any results.

Search published CPU2017 results

Last update: *2021-06-09t15:29*

| Search | CPU2017 Floating Point Rates | CPU2017 Floating Point Speed | CPU2017 Integer Rates | CPU2017 Integer Speed |

CPU2017 Floating Point Rates (6378):

[Search in CPU2017 Floating Point Rates results]

Test Sponsor	System Name	Base Copies	Processor			Results		Energy	
			Enabled Cores	Enabled Chips	Threads/ Core	Base	Peak	Base	Peak
ASUSTeK Computer Inc.	ASUS RS700-E9(Z11PP-D24) Server System (2.70 GHz, Intel Xeon Gold 6150) HTML | CSV | Text | PDF | PS | Config	72	36	2	2	199	201	--	--
ASUSTeK Computer Inc.	ASUS RS700-E9(Z11PP-D24) Server System (2.10 GHz, Intel Xeon Platinum 8176) HTML | CSV | Text | PDF | PS | Config	112	56	2	2	233	237	--	--
ASUSTeK Computer Inc.	ASUS RS700-E9(Z11PP-D24) Server System (2.70 GHz, Intel Xeon Gold 6150) HTML | CSV | Text | PDF | PS | Config	72	36	2	2	199	202	--	--
ASUSTeK Computer Inc.	ASUS RS700-E9(Z11PP-D24) Server System (2.10 GHz, Intel Xeon Platinum 8176) HTML | CSV | Text | PDF | PS | Config	112	56	2	2	233	236	--	--
ASUSTeK Computer Inc.	ASUS WS C621E SAGE Server System (2.50 GHz, Intel Xeon Platinum 8180) HTML | CSV | Text | PDF | PS | Config	112	56	2	2	252	257	--	--
ASUSTeK Computer Inc.	ASUS RS720Q-E9(Z11PH-D12) Server System (2.70 GHz, Intel Xeon Gold 6150) HTML | CSV | Text | PDF | PS | Config	72	36	2	2	205	209	--	--
ASUSTeK Computer Inc.	ASUS RS720Q-E9(Z11PH-D12) Server System (2.10 GHz, Intel Xeon Platinum 8176) HTML | CSV | Text | PDF | PS | Config	112	56	2	2	241	244	--	--
ASUSTeK Computer Inc.	ASUS RS720Q-E9(Z11PH-D12) Server System (3.50 GHz, Intel Xeon Gold 6144)								

8

(d) TPC Benchmarks

(*Transaction Performance Council*)

- SPEC's benchmarks are helpful to computer buyers whose principal concern is CPU performance.
- When the performance of the entire system under high transaction loads is a greater concern, the *TPC benchmarks* are more suitable.

- The current version of this suite is the *TPC-C benchmark*.
- TPC-C models the transactions typical of a **warehousing** and **distribution business using terminal emulation software**.

TPC-C Benchmark

from http://tpc.org/tpcc/results/tpcc_perf_results5.asp?resulttype=all

TPC-C Top Performance Results

Version 5 Results As of 13-Jun-2021 at 5:33 PM [GMT]

Note 1: The TPC believes it is not valid to compare prices or price/performance of results in different currencies.

(•) All Active Results ( ) Active Clustered Results ( ) Active Non-Clustered Results Currency: All [ ] Include Historical Results

Rank	Company	System	Performance (tpmC)	Price/tpmC	Watts/KtpmC	System Availability	Database	Operating System	TP Monitor	Date Submitted
1	ANT FINANCIAL	Alibaba Cloud Elastic Compute Service Cluster	707,351,007	3.98 CNY	NR	06/08/20	OceanBase v2.2 Enterprise Edition with Partitioning, Horizontal Scalability and Advanced Compression	Alibaba Aliyun Linux 2	Nginx 1.15.8	05/18/20
2	ANT FINANCIAL	Alibaba Cloud Elastic Compute Service Cluster	60,880,800	6.25 CNY	NR	10/02/19	OceanBase v2.2 Enterprise Edition with Partitioning, Horizontal Scalability and Advanced Compression	Alibaba Aliyun Linux 2	Nginx 1.15.8	10/01/19
3	TTA	AS-1124US-TNRP	380,475	654.00 KRW	NR	09/01/20	TTA Goldilocks v3.1 Standard Edition	Red Hat Enterprise Linux Server 7.7	JBoss Web Server 3.1	08/16/20
4	TTA	KTNF KR580S1	152,330	1,680.00 KRW	NR	11/20/18	TTA Goldilocks v3.1 Standard Edition	Red Hat Enterprise Linux Server 7.5	JBoss Web Server 3.1	11/19/18
5	TTA	KTNF KR570S1	152,328	1,592.00 KRW	NR	11/12/19	TTA Goldilocks v3.1 Standard Edition	Red Hat Enterprise Linux Server 7.7	JBoss Web Server 3.1	10/27/19

8 (e) System Simulation

- The TPC benchmarks differ from the SPEC benchmarks in that they try to simulate a complete computing environment.
- Although the purpose of the TPC benchmarks is to measure performance, their simulated environment may also be useful to predict, and hence optimize, performance under various conditions.

- In general, *simulations* give us tools that we can use to model and predict aspects of system behaviour without the use of the exact live environment that the simulator is *modeling*.

8.6 Summary

8

- Computer performance assessment relies upon measures of central tendency that include the *arithmetic mean*, *weighted arithmetic mean*, the *geometric mean*, and the *harmonic mean*.
- Each of these is applicable under different circumstances.
- *Benchmark suites* have been designed to provide objective performance assessment.
  - The most well respected of these are the *SPEC* and *TPC* benchmarks.

- *CPU performance* depends upon many factors.
  - These include pipelining, parallel execution units, integrated floating-point units, and effective branch prediction.
  - User code optimization affords the greatest opportunity for performance improvement.
  - Code optimization methods include loop manipulation and good algorithm design.

Review Questions

8

8.1 Which of the means is useful for comparing rates?

8.2 The execution times for three systems running five benchmarks are shown in the table below. Compare the relative performance of each of these systems (i.e., A to B, B to C, and A to C) using the arithmetic and geometric means. Are there any surprises? Explain.

Program	System A Execution Time	System B Execution Time	System C Execution Time
v	150	200	75
w	200	250	150
x	250	175	200
y	400	800	500
z	1000	1200	1100

Review Questions

8

8.3 The execution times for three systems running five benchmarks are shown in the table below. Compare the relative performance of each of these systems (i.e., A to B, B to C, and A to C) using the arithmetic and geometric means. Are there any surprises? Explain.

Program	System A Execution Time	System B Execution Time	System C Execution Time
v	45	125	75
w	300	275	350
x	250	100	200
y	400	300	500
z	800	1200	700