Answers to Exercises

♦1. Table 11.2 shows an execution mix and run times for two computers, System A and System C. In this example System C is 83% faster than System A. Table 11.3 shows run times for System A with a different execution mix. Using the execution mix in Table 11.3, calculate the percentage by which System C would be faster than System A. Using the original statistics from Table 11.2, by how much has the performance of System A degraded under the new execution mix?

Ans.

The weighted average execution time of System C is 2170 / 5 = 434. Thus, we have (563.5 / 434 = 1.298387 – 1) * 100 = 30%.

System A's performance has degraded by (9563.5 / 79 - 1) * 100 = 641.4%

---

2. With regard to the performance data cited for programs v, w, x, y, and z in Section 11.3, find the geometric means of the run times of the programs for System B and System C using System A as the reference system. Verify that the ratios of the means is consistent with the results obtained using the other two systems as reference systems.

Ans.

Geometric mean for B = 0.597604. Geometric mean for C = 0.412234.

---

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	150	200	75
w	200	250	150
x	250	175	200
y	400	800	500
z	1000	1200	1100

Ans.

System A				System B				System C			
150	1	1.3333	0.5	200	0.75	1	0.375	75	2	2.6667	1
200	1	1.25	0.75	250	0.8	1	0.6	150	1.3333	1.6667	1
250	1	0.7	0.8	175	1.4286	1	1.1429	200	1.25	0.875	1
400	1	2	1.25	800	0.5	1	0.625	500	0.8	1.6	1
1000	1	1.2	1.1	1200	0.8333	1	0.9167	1100	0.9091	1.0909	1
	Arithmetic mean				Arithmetic mean				Arithmetic mean		
	400				525				405		
	Geometric means				Geometric means				Geometric means		
	1	0.7712	1.1364		1.1596	1	1.3663		0.7946	0.6330	1

---

4. 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

Ans.

System A				System B				System C			
45	1	2.7778	1.6667	125	0.36	1	0.6	75	0.6	1.6667	1
300	1	0.9167	1.1667	275	1.0909	1	1.2727	350	0.8571	0.7857	1
250	1	0.4	0.8	100	2.5000	1	2.0000	200	1.25	0.5	1
400	1	0.75	1.25	300	1.3333	1	1.6667	500	0.8	0.6	1
800	1	1.5	0.875	1200	0.6667	1	0.5833	700	1.1429	1.7143	1
	Arithmetic mean				Arithmetic mean				Arithmetic mean		
	359				400				365		
	Geometric means				Geometric means				Geometric means		
	1	0.7881	0.8683		0.8402	1	0.8166		1.0753	0.9494	1

---

5. A company that is selling database management optimization software contacts you to pitch its product. The representative claims that the memory management software will reduce page fault rates for your system. She offers you a 30-day free trial of this software. Before you install it, however, you decide to first determine a baseline for your system. At specific times of the day, you sample and record the page fault rate of your system (using the system's diagnostic software). You do the same after the software has been installed. How much of an average performance improvement has the new software provided? (Hint: Use the harmonic mean.)

   The fault rates and times of day are shown in the table below.

Time	Fault Rate Before	Fault Rate After
02:00-03:00	35%	45%
10:00-11:00	42%	38%
13:00-14:00	12%	10%
18:00-19:00	20%	22%

Ans.

Before, the average rate is 21.5%. After, the rate is 21%. This gives approximately a 4% improvement.

---

6. What are the limitations of synthetic benchmarks such as Whetstone and Dhrystone? Do you think that the concept of a synthetic benchmark could be extended do overcome these limitations? Explain your answer.

Ans.

The main problem with Whetstone and Dhrystone is that most programs are so small. As such, each is easy to optimize and the entire program fits in cache memory on today's systems. Any single-function program will be subject to optimization tricks, so a synthetic benchmark must carry out a broad array of functions. The SPEC benchmarks are technically synthetic benchmarks, although they were derived from real applications.

♦ 7. What would you say to a vendor that tells you that his system runs 50% of the SPEC benchmark kernel programs twice as fast as the leading competitive system? Which statistical fallacy is at work here?

Ans.

This is the fallacy of incomplete information. The vendor is not telling you how his product compared to its competition on the other tests. The aggregate test results for both may be the same—or even worse for the system of the vendor with whom you are speaking.

---

8. Suppose that you are looking into purchasing a new computer system. You have suitable benchmark results for all of the systems that you are considering except for System X Model Q. The benchmark results have been reported for System X Model S, and they are not quite as good as several competing brands. In order to complete your research, you call System X computer company and ask when they plan to publish benchmark results for the Model Q. The company tells you that they will not be publishing these results anytime soon, but because the disk drives of Model Q give an average access time of 12ms, while Model S had 15ms drives, Model Q will perform better than Model S by 25%. How would you record the performance metrics for System X model Q?

Ans.

You don't record anything for System X Model Q. Amdahl's Law tells us that overall speedup is related to the extent to which a faster component is used. You have no way of knowing the extent to which the faster disk plays into the overall system equation. In fact, if the bus or cache components of the Model Q don't scale up to take advantage of the faster disk, there may be no performance improvement at all.

♦9. What value do you think there would be in comparing the results of two different SPEC CPU releases, say, SPEC95 with SPEC2000?

Ans.

There is no value in doing this. Each release consists of a different set of programs, so the results cannot be compared.

---

10. Besides the retail business sector, what other organizations would need good performance from a transaction-processing system. Justify your answer.

Ans.

Any business that requires a high volume of short transactions. Examples include airline reservation systems, banking systems, entertainment ticket sellers, and anyone that posts payments against invoices.

♦11. Which of the benchmarks discussed in this chapter would be most helpful to you if you were about to purchase a system to be used in DNA research? Why would you choose this one? Would any of the other benchmarks be of interest to you? Why or why not?

Ans.

The best benchmark in this situation would be the SPEC CPU series. In particular, the SPECRate metrics, since you'd be looking at a multiprocessor system. You may give some consideration to TPC-H or TPC-R, because enormous amounts of data are required in genome research.

---

12. Suppose a friend has asked you to help him to make a choice as to what kind of computer he should buy for his personal use at home. What would you look for in comparing various makes and models? How does your line of thinking differ in this situation than if you were to help your employer purchase a Web server to accept customers' orders over the Internet?

Ans.

In both cases, you should be mindful of manufacturers' warranties and service policies. For private use, comparisons are much simpler, as the architectures of these systems are very similar. Also, a private individual is probably more concerned about price than he is about performance. In considering a system that has economic value to a company, you should bring your knowledge of benchmarks (TPC-W and SPECWeb) to bear on the problem, along with various vendors' pricing policies.

♦13. Suppose you have just been assigned to a committee that has been tasked with purchasing a new enterprise file server that will support customer account activity as well as many administrative functions, such as producing a weekly payroll. (Yes, a committee frequently makes these decisions!) One of your committee members has just learned that a particular system has blown out the competition in the SPEC CPU2000 benchmarks. He is now insisting that the committee buy one of these systems. What would be your reaction to this?

Ans.

First, be diplomatic. Suggest that the group investigate whether TPC-C benchmarks are available for this system. Whether or not TPC-C figures are available, you may want to educate this individual as to the meaning of Amdahl's Law and why a fast CPU will not necessarily determine whether the entire system will handle your workload.

---

* 14. We discussed the limitations of the harmonic mean in its application to computer performance assessment. A number of critics have suggested that the SPEC should use the harmonic mean instead. Suggest a unit of "work" that would be appropriate for reformulating the SPEC benchmarks as rate metrics. Test your theory using results from SPEC's Web site, www.spec.org.

Ans.

Try using "lines of source code" as published in the benchmark. We must caution that "lines of code" is usually a poor metric of program complexity.

---

* 15. SPEC and the TPC both publish benchmarks for Web server systems. Visit the respective Web sites of these organizations (www.spec.org and www.tpc.org) to try to find identical (or comparable) systems that have results posted on both sites. Discuss your findings.

Ans.

No answer given.

---

16. We mentioned that a large volume of data is gathered during system probe traces. To give you some idea of the actual volume of data involved, suppose plans are being made to install a hardware probe that reports the contents of a system's program counter, instruction register, accumulator, memory address register, and memory buffer register. The system has a clock that runs at 1GHz. During each cycle of the system clock, the status of these 5 registers is written to nonvolatile memory attached to the probe circuitry. If each

register is 64 bits wide, how much storage will the probe require if it is to gather data for 2 seconds?

Ans.

$2 \times 2^{30} \times 2^5 \times 2^6 = 2^{42}$ bits

---

17. The sidebar in Section 11.5.2 presents ways in which program performance can be improved. For each of the tips in the sidebar, state whether a computer organization and architecture issue is involved. If so, explain the reasoning behind the advice as given. If you feel anything is missing from the sidebar, include your advice in your analysis.

Ans.

Give the compiler as much information as possible about what you are doing. Use constants and local variables where possible. If your language permits them, define prototypes, and declare static functions. Use arrays instead of pointers when you can.	Organization and architecture: Giving the compiler as much information as possible permits optimization. Arrays result in the allocation of contiguous memory. Thus, you can get some help from cache.
Avoid unnecessary type casting and minimize floating-point to integer conversions.	Organization and architecture: “On the fly” conversions result in “on the fly” storage allocations and consume machine cycles during the conversion.
Avoid overflow and underflow.	Organization and architecture: These situations cause exceptions that have to be dealt with at some level of the machine, thereby consuming machine cycles. Besides, if you have overflow or underflow, you should check your numerical algorithm.
Use a suitable data type (e.g., float, double, int).	Organization and architecture: As above, “on the fly” conversions can be avoided, as well as overflow and underflow.
Consider using multiplication instead of division.	Organization and architecture: Multiplication is generally faster, and there is no possibility of division by zero.
Eliminate all unnecessary branches.	Organization and architecture: Branching causes pipeline stalls that can cost several machine cycles.
Use iteration instead of recursion when possible.	Organization and architecture: Recursion is resource-intensive because parameters and a return address must be saved (and eventually restored) for each call of the function.
Build conditional statements (e.g., if, switch, case) with the most probable cases first.	Organization and architecture: We save machine cycles by testing fewer conditions.
Declare variables in a structure in order of size with the largest ones first.	Organization and architecture: Results in more efficient allocation of storage.
When a program is having performance problems, profile the program before beginning optimization procedures.	Human problem-solving.
Never discard an algorithm based solely on its original performance. A fair comparison can occur only when all algorithms are fully optimized.	Human problem-solving.

---

18. In our discussion of the physical aspects of disk performance, we stated that replacing 7200 RPM disks with 10,000 RPM disks can bring a 10% to 50% performance improvement. Why would an improvement of only 10% occur? How would it be that no improvement at all would occur? (Hint: Rotational latency is not the only determining factor of disk performance.)

Ans.

The rotational speed of a disk isn't the only determining factor of its performance. Seek time and transfer rate are also important. If the new disks have much lower seek times than the older ones, you could end up getting poorer performance out of those "faster" disks!

---

19. Calculate the number of disk tracks traversed using the FCFS, SSTF, SCAN, and LOOK algorithms for the series of disk track service requests given below. At the time the first request arrives in the disk request queue, the read/write head is at track 50, moving toward the outer (lower-numbered) tracks.

54, 36, 21, 74, 46, 35, 26, 67

Ans.

FCFS		SSTF		SCAN		LOOK	
Tracks	Distance	Tracks	Distance	Tracks	Distance	Tracks	Distance
50 → 54	4	50 → 46	4	50 → 46	4	50 → 46	4
54 → 36	18	46 → 54	8	46 → 36	10	46 → 36	10
36 → 21	15	54 → 67	13	36 → 35	1	36 → 35	1
21 → 74	53	67 → 74	7	35 → 26	9	35 → 26	9
74 → 46	28	74 → 36	38	26 → 21	5	26 → 21	5
46 → 35	11	36 → 35	1	21 → 0	21	21 → 54	33
35 → 26	9	35 → 26	9	0 → 54	54	54 → 67	13
26 → 67	41	26 → 21	5	54 → 67	13	67 → 74	7
				67 → 74	7		
Total	179	Total	85	Total	124	Total	82

---

20. Repeat the previous problem using tracks:

82, 97, 35, 75, 53, 47, 17, 11

Ans.

FCFS		SSTF		SCAN		LOOK	
Tracks	Distance	Tracks	Distance	Tracks	Distance	Tracks	Distance
50 → 82	32	50 → 47	3	50 → 47	3	50 → 47	3
82 → 97	15	47 → 53	6	47 → 35	12	47 → 35	12
97 → 35	62	53 → 35	18	35 → 17	18	35 → 17	18
35 → 75	40	35 → 17	18	17 → 11	6	17 → 11	6
75 → 53	22	17 → 11	6	11 → 0	11	11 → 53	22
53 → 47	6	11 → 75	64	0 → 53	53	53 → 75	22
47 → 17	30	75 → 82	7	53 → 75	22	75 → 82	7
17 → 11	6	82 → 97	15	75 → 82	7	82 → 97	15
				82 → 97	15		
Total	213	Total	137	Total	147	Total	105

---

21. On a particular brand of disk drive, the time that it takes for the disk arm to pass over a single disk track without stopping is 500 nanoseconds. However, once the head reaches the track for which it has a service request, it needs 2 milliseconds to "settle" over the required track before it can start reading or writing. Based on these timings, compare the relative times required for FCFS, SSTF, and LOOK to carry out the schedule given below. You will need to compare SSTF to FCFS, LOOK to FCFS, and LOOK to SSTF.

    As in our previous question, when the first request arrives in the disk request queue, the read/write head is at track 50, moving toward the outer (lower-numbered) tracks. The requested tracks are:

    35, 53, 90, 67, 79, 37, 76, 47

Ans.

FCFS		SSTF		SCAN		LOOK	
Tracks	Distance	Tracks	Distance	Tracks	Distance	Tracks	Distance
50 → 35	15	50 → 47	3	50 → 47	3	50 → 47	3
35 → 53	18	47 → 53	6	47 → 37	10	47 → 37	10
53 → 90	37	53 → 67	14	37 → 35	2	37 → 35	2
90 → 67	23	67 → 76	9	35 → 0	35	35 → 53	18
67 → 79	12	76 → 79	3	0 → 53	53	53 → 67	14
79 → 37	42	79 → 90	11	53 → 67	14	67 → 76	9
37 → 76	39	90 → 37	53	67 → 76	9	76 → 79	3
76 → 47	29	37 → 35	2	76 → 79	3	79 → 90	11
				79 → 90	11		
Total	215	Total	91	Total	140	Total	70

$\text{FCFS} = 215 \text{ tracks} \times 500\text{ns} + 8 \text{ stops} \times 2{\times}10^6 \text{ ns} = 16.1075 \text{ ms}$

$\text{SSTF} = 91 \text{ tracks} \times 500\text{ns} + 8 \text{ stops} \times 2{\times}10^6 \text{ ns} = 16.0455 \text{ ms}$

$\text{SCAN} = 140 \text{ tracks} \times 500\text{ns} + 8 \text{ stops} \times 2{\times}10^6 \text{ ns} = 16.07 \text{ ms}$

LOOK = 70 tracks × 500ns + 8 stops × 2×10^6 ns = 16.035 ms

---

22. Repeat Exercise 21 for disk tracks:

48, 14, 85, 35, 84, 61, 30, 22

Ans.

FCFS		SSTF		SCAN		LOOK	
Tracks	Distance	Tracks	Distance	Tracks	Distance	Tracks	Distance
50 → 48	2	50 → 48	2	50 → 48	2	50 → 48	2
48 → 14	34	48 → 35	13	48 → 35	13	48 → 35	13
14 → 85	71	35 → 30	5	35 → 30	5	35 → 30	5
85 → 35	50	30 → 22	8	30 → 22	8	30 → 22	8
35 → 84	49	22 → 14	8	22 → 14	8	22 → 14	8
84 → 61	23	14 → 61	47	14 → 0	14	14 → 61	47
61 → 30	31	61 → 84	23	0 → 61	61	61 → 84	23
30 → 22	8	84 → 85	1	61 → 84	23	84 → 85	1
				84 → 85	1		
Total	268	Total	107	Total	135	Total	107

FCFS = 268 tracks × 500 ns + 8 stops × 2×106 ns = 16.134 ms
SSTF = 107 tracks × 500 ns + 8 stops × 2×106 ns = 16.054 ms
SCAN = 135 tracks × 500 ns + 8 stops × 2×106 ns = 16.068 ms
LOOK = 107 tracks × 500 ns + 8 stops × 2×106 ns = 16.054 ms

---

* 23. In our discussion of the SSTF disk scheduling algorithm, we stated that the problem of starvation "is at its worst with low disk utilization rates." Explain why this is so.

Ans.

When the disk is busy, it is likely that many different tracks will be accessed. This increases the likelihood that the seeks will cover the entire disk. At low utilization, only a few tracks are used. If these tracks are distant from a seek request that has been waiting in the queue for "a long time," the nearer track seeks will continue to be serviced before the distant track request. Thus, we have starvation.

♦ 24. A certain microprocessor requires either 2, 3, 4, 8, or 12 machine cycles to perform various operations. Twenty-five percent of its instructions require 2 machine cycles, 20% require 3 machine cycles, 17.5% require 4 machine cycles, 12.5% require 8 machine cycles, and 25% require 12 machine cycles.

♦a) What is the average number of machine cycles per instruction for this microprocessor?
♦b) What is the clock rate (machine cycles per second) required for this microprocessor to be a "1 MIPS" processor?

♦ c) Suppose this system requires an extra 20 machine cycles to retrieve an operand from memory. It has to go to memory 40% of the time. What is the average number of machine cycles per instruction for this microprocessor including its memory fetch instructions?

Ans.

a) 0.25 * 2+ 0.2*3 + 0.175 * 4 + 0.125 * 8 + 0.25 * 12 = 5.8 cycles/instruction
b) 5.8 cycles/instruction * 1,000,000 instructions/second = 5.8 MHz
c) 5.8 * 0.6 + 25.8 * 0.4 = 13.8 cycles

---

25. A certain microprocessor requires either 2, 4, 8, 12, or 16 machine cycles to perform various operations. Seventeen and one-half (17.5) percent of its instructions require 2 machine cycles, 12.5% require 4 machine cycles, 35% require 8 machine cycles, 20% require 12 machine cycles, and 15% require 16 machine cycles.

a) What is the average number of machine cycles per instruction for this microprocessor?
b) What is the clock rate (machine cycles per second) required for this microprocessor to be a "1 MIPS" processor?
c) Suppose this system requires an extra 16 machine cycles to retrieve an operand from memory. It has to go to memory 30% of the time. What is the average number of machine cycles per instruction for this microprocessor including its memory fetch instructions?

Ans.

a) 2 * 0.175 + 4 * 0.125 + 8 * 0.35 + 12 * 0.2 + 16 * 0.15 = 8.45 cycles/instruction
b) 8.45 cycles/instruction * 1,000,000 instructions/second = 8.45 MHz
c) 8.45 * 0.7 + 24.45 *0.3 = 13.25 cycles

---

26. Herbert Grosch (b. 1918) has been an eminent computer scientist since the 1940s. In 1965, he put forth a claim that he "humbly" called Grosch's Law. This law can be paraphrased as:

> Computer performance increases as the square of the cost. If Computer A costs twice as much as Computer B, you should expect Computer A to be four times as fast as Computer B.

Use any of the TPC benchmarks to confirm or refute this claim. What happens when you restrict your comparison to similar systems? More specifically, check the price/performance statistics for systems running only the same class of operating system and database software. What happens when you don't restrict the comparison to similar systems? Do these results change when you select a different benchmark, e.g., TPC-W versus TPC-C? Discuss your findings.

Ans.

This is a research problem. It is recommended that the reader visit the TPC Web site to look at the price-performance statistics published there (as directed in the exercise). It is expected that the results will vary widely until the search is restricted to similar systems. For example, we should compare only "small Intel servers" with others like it.
