Problem 1

You are studying the performance of a program executing on a CPU which features a cache memory to speed up the execution of instructions. Every time the CPU executes an instruction, it can result in a cache "hit" or a cache "miss". Having a cache hit is beneficial to the overall runtime of the program. After studying the cache system, you have measured that an instruction results in a cache hit 30% of the time. This quantity is called *hit rate*. Assume that the hit rate never changes.

The program under analysis begins at the entry point; it then executes exactly 10 instructions. After that the execution might follow path A with probability 80%; or path B with probability 20%. When the program executes path A, it will execute 4 more instructions. After these, it will further execute path C with 6 more instructions or path D with 3 more instructions. From path A, execution will continue to path C with probability 30% and to path D with probability 70%. On path B, the program will execute 10 more instructions. Regardless of the path taken, the program will terminate at the end of path C, D, or B.

a) Provide the PMF for the total number of instructions that will be executed by the program, regardless of whether they result in cache hits or misses.

b) How many instructions will be executed on average across many runs of the program?

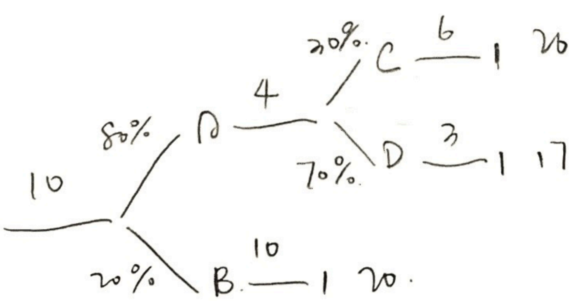
c) Assume that we want to observe one run in which the instructions on path are B being executed. How many times, on average, we should execute the program?

d) What is the probability that out of 30 consecutive executions of the program, path D will be executed 5 or more times?

10

e) What is the probability of observing exactly 10 cache hits in a generic run of the program?

f) What is the average number of hits you expect to see throughout the execution of the entire program?



(a). $P(X=20) = 0.8(0.3) + 0.2 = 0.44$

$P(X=17) = 0.8(0.7) = 0.56$.

$$P = \begin{cases} 0.44 & x=20 \\ 0.56 & x=17 \\ 0 & \text{otherwise.} \end{cases}$$

b). $0.44(20) + 0.56(17) = 18.32$

c). $P = 0.2$ Geometric Distribution

$\frac{1}{1-p} = \frac{1}{0.8} = 1.25$

(e). $p = 30\%$ $x = 10$. $n = 18.32$

$\binom{18.32}{10} 0.3^{10} (1-0.3)^{8.32} \approx 0.0133$

d). $P(C) = 0.8(0.3) = 0.24$ use binomial.

$n=30$ $x = 5$ or more. $P(X=0) = \binom{30}{0} 0.24^0 (1-0.24)^{30}$ $\boxed{= 2.657 \times 10^{-4}}$

$P(X=4) = \binom{30}{4} 0.24^4 (1-0.24)^{26} = 0.0724$

$P(X=3) = \binom{30}{3} 0.24^3 (1-0.24)^{27} = 0.0340$

$P(X=2) = \binom{30}{2} 0.24^2 (1-0.24)^{28} = 0.0115$

$P(X=1) = \binom{30}{1} 0.24 (1-0.24)^{29} = 0.0025$

$P(X \geq 5) = 1 - 0.0724 - 0.0340 - 0.0115 - 0.0025 - 2.657 \times 10^{-4} = 0.8793$.

f). $18.32(30\%) = 5.496$.

Problem 2.

$q = 23 \quad \lambda = 52 \quad T_s = 50ms = 0.05S$

a). T_q?

$q = \lambda \cdot T_q \Rightarrow 23 = 52 \cdot T_q \quad T_q = \frac{23}{52} = 0.4423S.$

b). $\frac{T_q}{T_s} = \frac{0.4423}{0.05} = 8.846.$

c). $\rho = \lambda \cdot T_s = 52(0.05) = 2.6$, so 3 CPUs

d). $\frac{2.6}{3} = 0.8667 \qquad$ e). $\frac{52}{3} = 17.3333$

f). $T_q = T_w + T_s \Rightarrow 0.4423 = T_w + 0.05 \quad T_w = 0.3923S$

g). Little's Law assumes a stable system so the arrival rate and departure rate are identical.
We also assume that the workload can be perfectly balanced among the CPUs

Problem 3.

a) $e^{\frac{-t}{MTBF}}$

$e^{\frac{-3}{2}} = 0.223$ $e^{\frac{-3}{5}} = 0.549$ $e^{\frac{-3}{10}} = 0.741$

0.223(0.96) = 0.214 0.549(0.90) = 0.494 0.741(0.85) = 0.630

so only M_3.

b). $P = 0.223$

use Geometric Distribution.

(1-p)/p = (1-0.214)/0.214 = 3.673

c). $n = 3$ $P_1 =$ 0.214 $P_2 =$ 0.494 $P_3 =$ 0.630 $X = 2$

use Binomial Distribution.

0.214 (1-0.494)(1-0.630) + 0.494 (1-0.214)(1-0.630) + 0.630 (1-0.214)(1-0.494) = 0.434

d). throughput = 18 * 0.3 *0.223 + 18 * 0.2 * 0.549 + 18 * 0.5 * 0.741 = 9.8496

e) Yes, I think the method is better than executing a request entirely on any of the individual machines, because it higher the total prob of correctly processed the requests, which is the same theory as putting eggs in to different baskets.