Exponential Averaging

Exponential Averaging is a CPU scheduling method that favors shortest CPU-burst first (SJF) scheduling. It is a weighted average that is used to predict and schedule the next CPU burst. The weighting is between the most recent CPU burst time, and all previous burst times. The exponential part is because the farther in the past a CPU burst time is, the less it contributes to the new prediction. This "aging" of past burst times is done by repeated multiplication of a weighting factor (repeated multiplication is exponential).

Here is the equation to use to generate predicted values:

$$PN+1 = ALP(TN) + (1 - ALP)PN$$

Where

PN+1 is the prediction for the next burst, used to schedule.
ALP is the weighting factor (between 0 and 1) that determines how quickly the past is "aged"
TN is the most recent CPU burst time
PN is the previous prediction used to schedule the most recent CPU burst.

For instance, a process with actual burst time (not known beforehand) of 5, 6, 1, 7 ... and a default prediction for the first burst of 10, with an alpha of 0.5

Pred	Actual	Calculation	Prediction for Next Burst
10	5	5 + 2.5	= 7.5
7.5	6	3.75 + 3	= 6.75
6.75	1	3.375 + 0.5	= 3.875
3.875	7	1.938 + 3.5	= 5.438

After each CPU burst, the predicted value would be calculated for the next burst. For instance, after the first two bursts of actual CPU time of 5 and 6, the prediction for the third burst would be 6.75. The 6.75 prediction would be used to schedule processes in order of the shortest predicted CPU burst first. This process's predicted CPU burst value that would be compared with the predicted burst of the other processes. The shortest would be executed next. If the 6.75 prediction was larger than other processes, there would all be scheduled to execute before this process gets CPU time.

Example

Calculate the exponential averaging with $T_1=10$, $\alpha=0.5$ and the algorithm is SJF with previous runs as $8, 7, 4, 16$.

a) 9
b) 8
c) 7.5
d) None

Use the Exponential Averaging Formula below to calculate the best answer;

$$S_{n+1}=a(T_n)+(1-a)*S_n$$

A possible interpretation of the problem is that we initially had jobs with actual run times 8, 7, 4, and 16.
Assuming no new jobs entered the queue, we'd schedule then in increasing order of their times, namely 4, 7, 8, 16 (since that's what SJF means--shortest job first).

Under these assumptions we'd have time slices

T1=10
T2??
T3??
T4??