Name: Kevin Zhang

Problem 1.

We want to design two sequences $b_1, b_2, ...$ and $s_1, s_2, ...$ such that for sufficiently large t, the two properties hold:

1. The values of stock and bond decreases over time: $b_1 \cdot b_2 \cdots b_t \leq 0.99^t$.
2. A strategy that rebalances across stock and bond after each day increases by at least 1.01^t.

A simple approach to accomplish this is to alternate gains. Suppose we have a large factor α and a small factor β. We can represent each sequence as follows:

$$b_1, b_2, b_3, ... = \alpha, \beta, \alpha, ...$$
$$s_1, s_2, s_3, ... = \beta, \alpha, \beta, ...$$

Then, for sufficiently large t, each individual sequence converges to $(\alpha\beta)^{t/2}$, but the overall gain for each day converges to $(\alpha+\beta)^t$. This can be seen because $b_1 \cdot b_2 = \alpha \cdot \beta$. Furthermore, $b_1 + s_1 = \alpha + \beta$. It then suffices to have the following hold:

$$\alpha \cdot \beta \leq 0.99^{1/2}$$
$$\alpha + \beta \geq 2.02$$

A simple example that satisfies this is $\alpha = 1.50$ and $\beta = 0.48$. The wealth growth of each day is 1.05, but each sequence decreases by 0.99.

Problem 2.

(a) Both the expert prediction and outcome is chosen uniformly randomly from $\{0, 1\}$. This means that for any given t, the probability that a prediction is correct is $1/2$. With 1 representing when a mistake was made and 0 representing when there wasn't a mistake, the expected number of mistakes for any given t is $1/2$. Thus:

$$\mathbb{E}[\text{\# of mistakes}] = \sum_{t=1}^{T} \mathbb{E}[\text{\# of mistakes at t}] = T/2$$

The expected number of mistakes for any algorithm is $T/2$.

(b) Let's examine the probability that there exists an expert with no mistake.

First, we can compute the probability that an indvidual expert i makes no mistakes:

$$\begin{aligned}
\Pr[\text{expert } i \text{ with no mistakes}] &= \prod_{t=1}^{T}(1/2) = (1/2)^T \\
&= 1/2^{logn - log(2lnn)} \\
&= 2^{log(2lnn) - logn} \\
&= \frac{2^{log(2lnn)}}{2^{logn}} \\
&\simeq \frac{2lnn}{n}
\end{aligned}$$

Then, we can express the probability no experts make no mistakes: (we use the fact that $1 - x \leq e^{-x}$)

$$\begin{aligned}
\Pr[\text{expert } i \text{ does not make no mistakes}] &= 1 - \frac{2lnn}{n} \\
\Pr[\text{no experts make no mistakes}] &= \prod_{i=1}^{n}(1 - \frac{2lnn}{n}) \\
&= (1 - \frac{2lnn}{n})^n \\
&\leq e^{-\frac{2lnn}{n} \cdot n} \\
&\leq e^{-2lnn} \\
&\leq \frac{1}{n^2}
\end{aligned}$$

Thus, the probability there exists an expert that makes no mistakes is at least $1 - \frac{1}{n^2}$. As such, with high probability, there is a best expert that makes 0 mistakes. Since any algo makes $T/2$ expected mistakes, with $T = logn - log(2lnn)$, with T being bounded by $logn$, any algorithm must make $\Omega(logn)$ more mistakes than the best expert.

Problem 3.

(a) We can express the probability that the number of zeros does not exceed $T/2 - \sqrt{T}/4$ as follows:

$$\Pr[\text{number of 0s is exactly } 0] = (1/2)^0 \times \binom{T}{0} \times (1/2)^T$$
$$\Pr[\text{number of 0s is exactly } 1] = (1/2)^1 \times \binom{T}{1} \times (1/2)^{T-1}$$
$$\Pr[\text{number of 0s is exactly } 2] = (1/2)^2 \times \binom{T}{2} \times (1/2)^{T-2}$$
$$\vdots$$
$$\Pr[\text{number of 0s is exactly } T/2 - \sqrt{T}/4] = (1/2)^{T/2-\sqrt{T}/4} \times \binom{T}{T/2 - \sqrt{T}/4} \times (1/2)^{T/2+\sqrt{T}/4}$$

$$\begin{aligned}
\Pr[\text{number of 0s does not exceed } T/2 - \sqrt{T}/4] &= \text{sum of the above} \\
&= \sum_{k=0}^{T/2-\sqrt{T}/4} (1/2)^k \times \binom{T}{k} \times (1/2)^{T-k} \\
&= \sum_{k=0}^{T/2-\sqrt{T}/4} (1/2)^T \times \binom{T}{k} \\
&= (1/2)^T \times \sum_{k=0}^{T/2-\sqrt{T}/4} \binom{T}{k}
\end{aligned}$$

(b) First, we want to show $\binom{T}{T/2} \leq \frac{2^T}{\sqrt{T}}$. This can be done using Stirling's formula ($n! \simeq (\frac{n}{e})^n \sqrt{2\pi n}$):

$$\begin{aligned}
\binom{T}{T/2} &= \frac{T!}{(T/2)!(T/2)!} \\
&\simeq \frac{(\frac{T}{e})^T \sqrt{2\pi T}}{[(\frac{T/2}{e})^{T/2}\sqrt{2\pi T/2}]^2} \\
&= \frac{(\frac{T}{e})^T \sqrt{2\pi T}}{[(\frac{T}{2e})^T \sqrt{\pi T}^2]} \\
&= \frac{2^T \sqrt{2\pi T}}{\sqrt{\pi T}^2} \\
&= \frac{2^T \sqrt{2}}{\sqrt{\pi T}} \\
&= \sqrt{\frac{2}{\pi}} \frac{2^T}{\sqrt{T}} \\
&\leq \frac{2^T}{\sqrt{T}}
\end{aligned}$$

Next, we want to use the above fact to show that the probability that the number of 0s does not exceed $T/2 - \sqrt{T}/4$ is at least $1/4$. To do so, there's a couple of useful tricks that we can use.

First, the probability that the number of 0s does not exceed $T/2 - \sqrt{T}/4$ is the same as the probability that the number of zeroes is greater than $T/2 + \sqrt{T}/4$. This is because we are in a binomial distributiion.

$$\Pr[\text{number of 0s does not exceed } T/2 - \sqrt{T}/4] = \Pr[\text{number of 0s exceeds } T/2 + \sqrt{T}/4]$$

$$(1/2)^T \times \sum_{k=0}^{T/2-\sqrt{T}/4} \binom{T}{k} = (1/2)^T \times \sum_{k=(T/2+\sqrt{T}/4)}^{T} \binom{T}{k}$$

Second, we can compute the probability as an *area under the curve* and approximate it using width and height computations. We can let width be $T/2 + \sqrt{T}/4$ and height be the highest point on the curve ($\frac{2^T}{\sqrt{T}}$). Thus:

$$
\begin{aligned}
(1/2)^T \times \sum_{k=(T/2+\sqrt{T}/4)}^{T} \binom{T}{k} &\simeq (1/2)^T (T/2 + \sqrt{T}/4)(\frac{2^T}{\sqrt{T}}) \\
&= (T/2)(\frac{1}{\sqrt{T}}) + (\sqrt{T}/4)(\frac{1}{\sqrt{T}}) \\
&= (\frac{\sqrt{T}}{\sqrt{2}}) + (\frac{1}{4}) \\
&\geq \frac{1}{4}
\end{aligned}
$$

$$
\Pr[\text{number of 0s does not exceed } T/2 - \sqrt{T}/4] \geq \frac{1}{4}
$$

Thus, with probability of at least $1/4$, the number of 0s does not exceed $T/2 - \sqrt{T}/4$.

Finally, we have shown that with constant probability, the number of 0s does not exceed $T/2 - \sqrt{T}/4$. Because our experts are optimistic and pessimistic, the better expert will simply be T minus the mistakes made by the worse expert. As such, the expected number of mistakes of the best expert is $T/2 - \sqrt{T}/4$. Since any algorithm has an expected number of mistakes of $T/2$, we have a $\Omega(\sqrt{T})$ bound.

Problem 4.

Let's consider a more adaptive version of the weighted majority algorithm, such that a weight $w_i^{(t)}$ only decreases when $w_i^{(t-1)} \geq \frac{\varepsilon\phi^{(t-1)}}{n(1-\varepsilon)}$. We want to find the number of mistakes the algorithm makes step T_1 to T_2 compared to an expert i. Let M^T be then number of mistakes the algorithm makes, and $m_i^{(T)}$ be the number of mistakes from the expert.

We can examine how $\phi^{(t)}$ changes. For each weight, we have three possibilities:

1. $w_i^{(t)} = w_i^{(t-1)}$ for when expert i is right.
2. $w_i^{(t)} = w_i^{(t-1)}$ for when expert i is wrong, but $w_i^{(t-1)} < \frac{\varepsilon\phi^{(t-1)}}{n(1-\varepsilon)}$.
3. $w_i^{(t)} = (1-\varepsilon)w_i^{(t-1)}$ for when expert i is wrong, and $w_i^{(t-1)} \geq \frac{\varepsilon\phi^{(t-1)}}{n(1-\varepsilon)}$.

If we substitute for the last condition, we get $w_i^t \geq \frac{\varepsilon}{n}\phi^{(t-1)}$. Since $\phi^{(t)}$ is the sum of weights, and the weights are decreasing by at most $\frac{\varepsilon}{n}$, we get the following:

$$\phi^{(t)} \leq (1-\frac{\varepsilon}{n})\phi^{(t-1)}$$

By induction over the interval T_1 to T_2, we get:

$$\phi^{(T_2)} \leq (1-\frac{\varepsilon}{n})^{M^T}\phi^{(T_1)}$$

We can also find a lower bound for $\phi^{(T_2)}$: this is the case of an expert having their weight reduced for every iteration.

$$\phi^{(T_2)} \geq (1-\varepsilon)^{m_i^{(T)}}\phi^{(T_1)}$$

Combining the two inequalities, we get:

$$\begin{aligned}
(1-\varepsilon)^{m_i^{(T)}}\phi^{(T_1)} &\leq (1-\frac{\varepsilon}{n})^{M^T}\phi^{(T_1)} \\
(1-\varepsilon)^{m_i^{(T)}} &\leq (1-\frac{\varepsilon}{n})^{M^T} \\
m_i^{(T)}\ln(1-\varepsilon) &\leq M^T\ln(1-\frac{\varepsilon}{n}) \\
M^T &\geq m_i^{(T)}\frac{\ln(1-\varepsilon)}{\ln(1-\frac{\varepsilon}{n})}
\end{aligned}$$