Exercise 1 (Yao’s principle). We are running an hard-drive and we want to minimize its energy consumption. When the hard drive is on, it consumes $x$ per unit of time; when it is off, it cost $y$ to turn back on. Request to access to the hard drive arrives at unknown dates $t_1 < t_2 < \cdots$ and we want to minimize the overall cost.

▶ Question 1.1) Show that we can restrict the study to minimize the energy consumption during each time interval $[t_i, t_{i+1}]$ independently and thus restrict ourselves to the following situation: at time $t = 0$, the drive is on and we want to serve a request arriving at an unknown beforehand date $t$ for a minimum cost.

Answer. $\triangleright$ Since each request is served as soon as it arrives, the drive has to be on at every time $t_i$. The energy consumption in an optimal solution has to be minimal in every time interval $[t_i, t_{i+1}]$. We are thus left with finding the best strategy to have the drive on at time $t_{i+1}$ given that it is on at time $t_i$. Since the cost paid for time interval $[t_i, t_{i+1}]$ is insensitive to time-translation, we may as well assume that $t_i = 0$ and that the request arrives at time $t = t_{i+1} - t_i$ (note that if the drive is off at time $0$, it is optimal to leave it off until the first request arrives at time $t_1$ and then pay only $y$; so we may assume that the drive is on at time $0$). $\triangleright$

▶ Question 1.2) Show that the only deterministic algorithms to consider are $A_d$, for $d \geq 0$: if the request arrives before $t = d$, serves it; otherwise, turn off the drive at time $d$ and turn it back on when the request arrives.

Answer. $\triangleright$ The only actions available to the algorithm are shutting on and off the hard drive. If the algorithm decides to shut down the hard drive after some wait (recall that the algorithm is informed of the arrival of the request at its arrival and no earlier), the algorithm will have to pay $y$ to shut it back on anyway and thus should not shut it back on before the request arrives in order to avoid consuming energy before the request arrives. Formally, any strategy that decides to shut off the hard drive after waiting $d$ seconds and shuts it back on when the request arrives, consumes strictly less than any other strategy that shuts off the drive after waiting $d$ seconds. Since we consider deterministic strategies and since the algorithm knows the arrival time of the requests at its arrival at time $t$ and no earlier, the wait of the algorithm before it shuts off the drive cannot depend on $t$ and has to be fixed. We conclude that any deterministic algorithm consumes at least as much as algorithm $A_d$ for some $d$. $\triangleright$

We denote by $I_t$ the instance where the request arrive at time $t$ and by $\text{OPT}(I_t)$ the optimal energy consumption to serve a request arriving at time $t$.

▶ Question 1.3) Show that:

$$\text{cost}(A_d(I_t)) = \begin{cases} xt & \text{if } t \leq d \\ xd + y & \text{if } t > d \end{cases} \quad \text{and} \quad \text{OPT}(I_t) = \min(xt, y).$$

Answer. $\triangleright$ As proved earlier, the only algorithm to consider are the one that shut the drive off and non at most once before the request arrives. If the algorithm decides to shut the drive off, it should do it at time $0$ to avoid wasting energy. The only two possible optimal strategies are thus: either letting the drive on until the request arrives, which costs $xt$; or shutting the drive off at time $0$ and pay $y$ at time $t$ to shut it back on when the request arrives. Thus, $\text{OPT} = \min(xt, y)$.

Now, consider algorithm $A_d$. If $t \leq d$, then the algorithm leaves the drive on until $t$ and pays $xt$. If $t > d$, then the algorithm waits $d$ units of time and pays $xd$, shuts the drive off and pays $y$ to shut it back on at time $t$. Thus $\text{cost}(A_d(I_{t\leq d})) = xt$ and $\text{cost}(A_d(I_{t>d})) = xd + y$. $\triangleright$

We define the competitive ratio of $A_d$ as: $C_d = \max_{t \geq 0} \frac{\text{cost}(A_d(I_t))}{\text{OPT}(I_t)}$. 

1
\begin{question}
Show that the expected cost paid by the line player to column player is:
\end{question}

\begin{answer}
Imagine that the date of arrival of the request is chosen by some malicious adversary trying to make the algorithm $A_d$ look as bad as possible with respect to the optimal strategy. First remark that once the algorithm has shut down the hard drive, it will pay no more than $xd + y$ so the adversary should not send the request long after time $d$ since only the optimal cost may increase after time $d$. The adversary should not send the request before $d$ either, because before $d$ the price paid by the algorithm increases at its maximal rate $x$ which is at least as large as the rate at which the optimum increases ($x > 0$). The worst competitive ratio of $A_d$ is thus obtained at time $d + \epsilon$ for $\epsilon \to 0$:

$$C_d = \frac{xd + y}{\min(xd, y)}.$$Clearly, $C_d > 2$ when $xd \neq y$ since $xd + y > 2 \min(xd, y)$; and $C_{y/x} = 2$. The optimal deterministic strategy is thus $C_{y/x}$.

Note, we can also obtain the same result by calculation: if $t \leq d$, then

$$\frac{\text{cost}(A_d(t))}{\text{OPT}(t)} = \frac{xt}{\min(xt, y)} = \max(1, xt/y)$$which is maximum for $t = d$ with value $\max(1, xd/y)$; if $t > d$,

$$\frac{\text{cost}(A_d(t))}{\text{OPT}(t)} = \frac{xd + y}{\min(xt, y)} = \max((d + \frac{y}{x})/t, 1 + \frac{xd}{y})$$which is maximum for $t \to d$ with value $1 + \max(\frac{y}{x}, \frac{xd}{y})$. It follows that $C_d = 1 + \max(\frac{y}{x}, \frac{xd}{y})$ which is minimum for $y = xd$.
\end{answer}

We turn now to randomized algorithms. Any randomized algorithm can be seen as a distribution $p : \mathbb{R} \to \mathbb{R}$ over deterministic algorithms: $A_p$ is the algorithm that runs $A_d$ "with probability $p(\delta)$" (more precisely, $A_p$ runs algorithm $A_\delta$ with $\delta \geq d$ with probability $\int_d^\infty p(\delta)d\delta$). Then,

$$\mathbb{E}[\text{cost}(A_p(I_t))] = \int_0^\infty \text{cost}(A_\delta(I_t))p(\delta)d\delta$$and the randomized competitive ratio is

$$C_p = \max_{t \geq 0} \frac{\mathbb{E}_p[\text{cost}(A_p(I_t))]}{\text{OPT}(I_t)}.$$Our goal is to find an optimal distribution $p^* = \arg\min_{p: \mathbb{R}^+ \to \mathbb{R}^+} \int_0^\infty \text{cost}(A_\delta(I_t))p(\delta)d\delta = \max_{t \geq 0} \frac{\mathbb{E}_p[\text{cost}(A_p(I_t))]}{\min(xt, y)}$. Note that we assume that the adversary choosing the instance $t$ does not known our random choice, otherwise we would be back to the deterministic case.

\textbf{Zero-sum games.} Zero-sum games are defined by a cost matrix $C = (c_{d,t})_{d=1..D, t=1..T}$. At each round, the line player chooses a line $d$ and the column player chooses (independently) a column $t$ and the line player pays $c_{d,t}$ to the column player (each player earns what the other looses). Now, an optimal deterministic strategy for the line player is a line $d^* \in \arg\min_d \max_t c_{d,t}$ whereas an optimal deterministic strategy for the column player is a column $t^* \in \arg\max_t \min_d c_{d,t}$. It turns out that optimal deterministic strategies do not always exist.

Let us focus on randomized strategies: assume that the line player chooses a line $d \in \{1, \ldots, D\}$ with probability $p_d$ and that the column player chooses a column $t \in \{1, \ldots, R\}$ with probability $q_t$. Let us see $p$ and $q$ as line and column vectors of dimension $D$ and $T$ respectively.

\begin{question}
Show that the expected cost paid by the line player to column player is:
\end{question}

\begin{answer}
When the row player chooses row $d$ and the other player chooses column $t$, the row player pays $c_{d,t}$. Since these two events are independent and of probability $p_d$ and $q_t$ respectively, the expected cost is

$$\sum_{d=1}^D \sum_{t=1}^T p_d c_{d,t} q_t = p^T C^T q.$$An optimal randomized strategy $p^*$ for the line player must minimize the worst cost to the worst possible distribution of the column player:

$$p^* = \arg\min_{p \geq 0} \max_{q \geq 0} p^T C q$$
whereas an optimal randomized strategy \( q^* \) for the column player must maximize the cost to the best possible distribution of the line player:

\[
q^* = \arg \max_{q \geq 0; \|q\|_1 = 1} \min_{p \geq 0; \|p\|_1 = 1} p C q
\]

Von Neumann showed that there always exists an optimal randomized strategy for both players, and the cost of these two strategies always match! (the minimax theorem is in fact equivalent to the duality theorem in linear programming):

\[
\min_{p \geq 0; \|p\|_1 = 1} \max_{q \geq 0; \|q\|_1 = 1} p C q = \max_{q \geq 0; \|q\|_1 = 1} \min_{p \geq 0; \|p\|_1 = 1} p C q
\]

**Question 1.6** Show the easy inequalities:

\[
\min_p \max_q p C q = \min_p \max_q C_d q = \min_p \min_q p C q
\]

where \( C_d \) and \( C_t \) denote respectively the \( d \)-th row and the \( t \)-th column of the cost matrix.

**Answer.** Since the first quadrant of the \( l_1 \)-sphere is compact in finite dimension and

the function \( \Gamma_p : q \mapsto p C q \) is continuous, there is a \( q' \) such that \( p C q \leq p C q' \) for all \( q \geq 0 \) such that \( \|q\|_1 = 1 \). Since \( q' \) is a probability distribution over \( \{1, \ldots, T\} \), \( (p C) q' = \sum_{t=1}^{T} q'_t \cdot (p C_t) \) is a convex combination of the values \( p C_t \) and is thus maximum when all the weight of \( q' \) is put on the maximum value, i.e., for \( q'_t = 1 \) for all \( t \) such that \( \arg \max_t p C_t \). It follows that \( \max_p p C q = \max_t p C_t \) for all \( p \) and similarly, \( \min_d C_d q = \min_p p C q \) for all \( q \). This demonstrates the two equalities.

By compactness and continuity of \( p \mapsto \max_p p C q \) and \( q \mapsto \min_p p C q \), let \( p^* \geq 0 \) with \( \|p^*\|_1 = 1 \) such that \( \max_p p C q \) is minimum and let \( q^* \geq 0 \) with \( \|q^*\|_1 = 1 \) which maximizes \( \min_p p C q^* \). Then, \( \min_p \max_q p C q = \max_q \min_p p C q^* \geq \max_p p C q^* \geq \min_p p C q = \max_q \min_p p C q \).

**Yao’s principle.** Yao remarked that one can see the interaction between a randomized algorithm and an adversary as a zero-sum game where at each round, the algorithm chooses a time \( d \) and the adversary chooses (simultaneously) a time \( t \) and the algorithm pays \( c_{d,t} = \frac{\text{cost}(A_d(I_t))}{\text{OPT}(I_t)} \) to the adversary. It follows from von Neumann’s minimax extended to our continuous setting:

\[
\min_{p \in \mathcal{D}} \max_{q \in \mathcal{Q}} \mathbb{E}_p \left[ \frac{\text{cost}(A_p(I_t))}{\text{OPT}(I_t)} \right] = \max_{q \in \mathcal{Q}} \min_{p \in \mathcal{D}} \mathbb{E}_q \left[ \frac{\text{cost}(A_d(I_q))}{\text{OPT}(I_q)} \right]
\]

where \( \mathcal{D} = \{ f : \mathbb{R}_+ \to \mathbb{R}_+ : \int_0^{\infty} f(x) dx = 1 \} \). That is to say “the expected cost of the best randomized algorithm on the worst instance is equal to the expected cost of the best deterministic algorithm for the worst distribution of instances”. The weak version proved in Question 1.6 says that “the expected cost of any randomized algorithm on the worst instance is always at least the expected cost of the best deterministic algorithm for the worst distribution of instances”.

We can now use this principle to prove that some strategies are optimal: if we can find two probability distribution \( p^* \) and \( q^* \) for \( d \) and \( t \) respectively for which the worst competitive ratio of \( A_{p^*} \) is equal to the best competitive ratio of a deterministic algorithm for distribution \( q^* \), then \( A_{q^*} \) is an optimal randomized algorithm.

Rescale so that \( x = y = 1 \) and consider the two distributions:

\[
p^*(d) = \begin{cases} \frac{e^d}{e-1} & \text{if } 0 \leq d \leq 1 \\ 0 & \text{otherwise} \end{cases}
\]

and

\[
q^*(t) = \begin{cases} \frac{e^{-t}}{e-1} & \text{if } 0 \leq t \leq 1 \\ 0 & \text{if } t > 1 \text{ and } t \neq 2 \end{cases}
\]

with a probability mass \( \text{Pr}\{ t = 2 \} = \frac{1}{e-1} \).
**Question 1.7)** Show that the expected competitive ratio of the randomized algorithm $A_{p^*}$ is independent of $t$ and that the expected competitive ratio for distribution of instance $q^*$ is independent of the deterministic algorithm and conclude that $A_{p^*}$ is optimal. What is the best expected competitive ratio achievable by a randomized algorithm?

**Answer.** Recall that drawing $d$ and $t$ according to the continuous probability distributions $p^*$ and $q^*$ respectively mean that $d \in [\delta, \delta + d\delta]$ with probability $p^*(\delta)d\delta$ and $t \in [\tau, \tau + dt]$ with probability $q^*(\tau)d\tau$. Let us first verify that $p^*$ and $q^*$ are valid probability distributions:

\[
\int_0^\infty p^*(\delta)d\delta = \int_0^1 e^{\delta} e^{-1} d\delta = 1, \text{ and} \\
\int_0^\infty q^*(t)dt = \frac{e}{e-1} \int_0^1 te^{-t}dt + \frac{1}{e-1} = \frac{e}{e-1} \left( [-e^{-t}]_0^1 - \int_0^1 -e^{-t}dt \right) + \frac{1}{e-1} = \frac{e(-1/e-1/e+1)+1}{e-1} = 1
\]

By definition, $A_{p^*}$ chooses with probability $p^*(\delta)d\delta$ to wait until some time $d \in [\delta, \delta + d\delta]$. Thus, given a request arriving at time $t$, the expected competitive ratio $C_{p^*, t}$ of $A_{p^*}$ is:

\[
C_{p^*, t} = \mathbb{E}_{d \sim p^*} \left[ \frac{\text{cost}(A_d(I))}{\text{OPT}(I)} \right] = \frac{\mathbb{E}_{d \sim p^*} [\text{cost}(A_d(I))]}{\min(t, 1)} = \frac{\int_0^t \text{cost}(A_{d \leq t}(I)) p^*(\delta)d\delta + \int_t^\infty \text{cost}(A_{d > t}(I)) p^*(\delta)d\delta}{\min(1, t)}
\]

Assume that $t \leq 1$, then:

\[
C_{p^*, t} = \frac{1}{t} \int_0^t (\delta + 1)e^{\delta} e^{-1}d\delta + \frac{1}{e-1} \int_t^1 e^{\delta} e^{-1}d\delta = \frac{1}{t(e-1)} \int_0^t d(\delta e^\delta) + \frac{1}{e-1} \int_t^1 d(e^\delta) = \frac{e^t}{e-1} + \frac{e-e^t}{e-1} = \frac{e}{e-1}.
\]

Assume now that $t \geq 1$, then: (note that this calculation is useless since the prices paid by the algorithm and the optimal remain unchanged when $t > 1 = y/x \geq d$)

\[
C_{p^*, t} = \int_0^1 \frac{(\delta + 1)e^{\delta}}{e-1}d\delta = \frac{1}{e-1} \int_0^1 d(\delta e^\delta) = \frac{e}{e-1}.
\]

Let us now estimate the expected competitive ratio $C_d(I_{q^*})$ of a given deterministic algorithm $A_d$ for some $d$ when the request arrival date follows the law $q^*$:

\[
C_{d, q^*} = \mathbb{E}_{t \sim q^*} \left[ \frac{\text{cost}(A_d(I))}{\text{OPT}(I)} \right] = \mathbb{E}_{t \sim q^*} \left[ \frac{\text{cost}(A_d(I))}{t} \right], \text{ since } \Pr_{t \sim q^*}\{t \leq 1\} = 1
\]

\[
= \int_0^d \frac{t}{d} q^*(t)dt + \int_d^\infty d + 1 \frac{1}{t} q^*(t)dt = \frac{e}{e-1} \int_0^{\min(d, 1)} te^{-t}dt + \frac{e}{e-1} \int_{\min(1, d)}^1 (d+1)e^{-t}dt + \eta,
\]

where $\eta = \frac{d+1}{e-1}$ if $d < 2$ and $\eta = \frac{2}{e-1}$ otherwise. Assume first that $d \leq 1$, then:

\[
C_{d, q^*} = \frac{e}{e-1} \left( [-e^{-t}]_0^d - \int_0^d -e^{-t}dt \right) + \frac{e(d+1)(e^{-d}-1/e)}{e-1} + \frac{1}{e-1} = \frac{-de^{1-d} - e^{1-d} + (d+1)e^{1-d} - (d+1) + d + 1}{e-1} = \frac{e}{e-1}
\]
Consider now $1 < d < 2$, then:

$$C_{d,q^*} = \frac{e}{e - 1} \left( \left[-te^{-t}\right]_0^1 - \int_0^1 e^{-t} dt \right) + \frac{d + 1}{e - 1} = -1 - 1 + e + d + 1 \frac{e}{e - 1} = \frac{e - 1 + d}{e - 1} > \frac{e}{e - 1}.$$  

Finally, if $d \geq 2$, then

$$C_{d,q^*} = \frac{e}{e - 1} \left( \left[-te^{-t}\right]_0^1 - \int_0^1 e^{-t} dt \right) + 2 \frac{e}{e - 1} = \frac{-1 - 1 + e + 2}{e - 1} = \frac{e}{e - 1}.$$  

$\triangleright$

It follows that $\max_t C_{p^*,t} = \frac{e}{e - 1} = \min_d C_{d,q^*}$ which implies by Yao’s principle that $\frac{e}{e - 1}$ is the best expected competitive ratio achievable by a randomized algorithm and it is obtained by the randomized algorithm $A_{p^*}$ which chooses the maximum waiting time $d$ according to law $p^*$.  

\[\]