Homework 1

1. **[6 points]** Consider the following weighted version of non-preemptive bipartite matching in the vertex arrival setting. Each offline vertex i has a non-negative weight w_i, and these weights are provided in the beginning with the offline vertices. The weight of a matching is the sum of the weights of the matched offline vertices, and we want to compete with the weight of the maximum weight matching. The rest of the problem definition is the same: in each round, a new online vertex appears along with the edges incident on it, and we must match it irrevocably to some available neighbor or leave it unmatched. Consider the following algorithm.

   - For each offline vertex i, sample $X_i \sim U[0,1]$ independently. ($U[0,1]$ denotes the uniform distribution on $[0,1]$.)
   - For each online vertex j, if j has at least one unmatched neighbor, match j to that unmatched neighbor i which maximizes $w_i \cdot (1 - e^{X_i - 1})$.

   Prove that this algorithm is $(1-1/e)$-competitive. (Observe that when the weights w_i are all equal, this algorithm is same as the Karp-Vazirani-Vazirani algorithm. The analysis will almost follow the same footsteps, but be careful and spot all the differences. If the proof of some claim in your analysis is exactly the same as in the Karp-Vazirani-Vazirani analysis, you may say so and skip the proof.)

   Answer: Define

$$\mathrm{rev}_i = \begin{cases} w_i e^{X_i - 1}, & \text{if item } i \text{ sold,} \\ 0 & \text{otherwise,} \end{cases} \tag{1}$$

$$\mathrm{util}_j = \begin{cases} w_i \cdot (1 - e^{X_i - 1}), & \text{if buyer } j \text{ buys item } i, \\ 0 & \text{if buyer } j \text{ doesn't buy any item.} \end{cases} \tag{2}$$

   As in the unweighted case, $\mathrm{ALG} = \sum_{\text{item } i} \mathrm{rev}_i + \sum_{\text{buyer } j} \mathrm{util}_j$. Now in this weighted case, we claim

$$\forall \text{ edge } (i,j)\text{: } \mathbb{E}[\mathrm{rev}_i] + \mathbb{E}[\mathrm{util}_j] \geq (1 - 1/e) \cdot w_i,$$

   after which the proof follows as in the unweighted case. Like in the unweighted case, we prove this claim conditional on every possible price of items other than i, and the expectation is taken over the price of i only. Define the market G' exactly as in the unweighted case, and define rev'_i and util'_j for G' in a manner analogous to equations (1) and (2) respectively. Define the sets A_l, A'_l as in the unweighted case. Then the claim that $A'_l \subseteq A_l$ and $|A_l| \leq |A'_l| + 1$ continues to hold. This implies that $\mathrm{util}_j \geq \mathrm{util}'_j$ with probability 1, and therefore, $\mathbb{E}_{X_i}[\mathrm{util}_j | P_{-i} = p_{-i}] \geq \mathrm{util}'_j(p_{-i})$. Henceforth, for succinctness, we define $u = \mathrm{util}'_j(p_{-i})$.

   Next, analogous to the unweighted case, one can prove that item i is sold (in G) if $w_i(1 - e^{X_i - 1}) > u$. Here you are bound to make the following **mistake** if you are not careful.

   > The condition $w_i(1 - e^{X_i - 1}) > u$ is equivalent to
   >
   > $$X_i < 1 + \ln\left(1 - \frac{u}{w_i}\right).$$
   >
   > Thus,
   >
   > $$\mathbb{E}_{X_i}[\mathrm{rev}_i | P_{-i} = p_{-i}] \geq \int_0^{1+\ln\left(1-\frac{u}{w_i}\right)} w_i e^{x-1} dx = w_i e^{x-1}\Big|_0^{1+\ln\left(1-\frac{u}{w_i}\right)} = w_i - u - \frac{w_i}{e},$$
   >
   > and therefore,
   >
   > $$\mathbb{E}_{X_i}[\mathrm{util}_j | P_{-i} = p_{-i}] + \mathbb{E}_{X_i}[\mathrm{rev}_i | P_{-i} = p_{-i}] \geq w_i - \frac{w_i}{e},$$
   >
   > as required.

This is a mistake because w_i could be greater than u, in which case, $\ln(1 - u/w_i)$ is meaningless. The fix is to consider two cases: $u < w_i \cdot (1 - 1/e)$ and $u \geq w_i \cdot (1 - 1/e)$. In the former case, the argument in the box holds. In the latter case, $\mathbb{E}_{X_i}[\text{util}_j | P_{-i} = p_{-i}] \geq \text{util}'_j(p_{-i}) = u$ itself is at least $w_i \cdot (1 - 1/e)$, so $\mathbb{E}_{X_i}[\text{util}_j | P_{-i} = p_{-i}] + \mathbb{E}_{X_i}[\text{rev}_i | P_{-i} = p_{-i}] \geq w_i \cdot (1 - 1/e)$ holds trivially, since rev_i is non-negative.

2. [**4 points**] In the third recorded lecture around the 48:00 timestamp (on the third page of the scanned notes), I wrote the following claim:

$$\mathbb{E}[|M|] \leq n\left(1 - \frac{1}{e}\right) + o(n),$$

and left the details for you to figure out. Complete that proof. (Again, you don't have to reprove the claims already proven in the recorded lecture.)

Answer: Let n' be the minimum number such that

$$\sum_{j=1}^{n'} \frac{1}{n - j + 1} \geq 1.$$

We use the following bounds.

$$\sum_{j=1}^{i} \Pr[(r_i, v_j) \in M] \leq \begin{cases} \sum_{j=1}^{i} \frac{1}{n-j+1} & \text{for } i < n', \\ \sum_{j=1}^{n'} \frac{1}{n-j+1} & \text{for } i \geq n', \end{cases}$$

that is,

$$\sum_{j=1}^{i} \Pr[(r_i, v_j) \in M] \leq \sum_{j=1}^{\min(i,n')} \frac{1}{n - j + 1}.$$

Thus,

$$\mathbb{E}[|M|] \leq \sum_{i=1}^{n} \sum_{j=1}^{\min(i,n')} \frac{1}{n - j + 1} = \sum_{j=1}^{n'} \sum_{i=j}^{n} \frac{1}{n - j + 1} = \sum_{j=1}^{n'} \frac{1}{n - j + 1} \sum_{i=j}^{n} 1 = \sum_{j=1}^{n'} 1 = n'.$$

Now let us bound n' from above. We have,

$$1 \geq \sum_{j=1}^{n'-1} \frac{1}{n - j + 1} \geq \int_{n-n'+2}^{n+1} \frac{1}{x} \cdot dx = \ln\left(\frac{n+1}{n - n' + 2}\right),$$

which imples,

$$\mathbb{E}[|M|] \leq n' \leq (n+1) \cdot \left(1 - \frac{1}{e}\right) + 1 = n \cdot \left(1 - \frac{1}{e}\right) + \left(2 - \frac{1}{e}\right),$$

as required.

Homework 2

In the *real-time secretary* problem, numbers from an adversarially chosen set $\{x_1, \ldots, x_n\}$ appear at their respective arrival times $T_1, \ldots, T_n$ that are distributed independently and uniformly in $[0, 1]$. As usual, whenever a number appears, an algorithm is allowed to either pick it and discard the remaining input, or discard it and continue. Like in the secretary problem, our goal is to design an algorithm that maximizes the probability of "success"; we say that the algorithm succeeds if it picks $\max_i x_i$. Crucially, in this case, n is not known to the algorithm in advance (otherwise it could simply behave like the secretary algorithm, because the numbers appear in a uniformly random order).

1. **[6 points]** Consider the following algorithm for real-time secretary, which involves a parameter $\tau \in [0, 1]$.

   - Discard all numbers arriving before time τ. Let θ be the maximum of all numbers that arrive before time τ. ($\theta = -\infty$ if no number appears before time τ.)
   - Thereafter, accept the earliest arriving number which exceeds θ.

   Derive a lower bound on the success probability of this algorithm as a function of τ. Hence show that there exists a τ for which the success probability is at least $1/e$.

   Answer: Let the random variable T^* denote the arrival time of $\max_i x_i$, and let the random variable $\mathcal{S}$ denote the set $\{i : T_i < T^*\}$. Then we have for $t < \tau$,

   $$\Pr[\text{success}|T^* = t] = 0.$$

   For $t \geq \tau$, we have,

   $$\Pr[\text{success}|T^* = t, \mathcal{S} = \emptyset] = 1.$$

   For any subset S of indices not containing $\arg\max_i x_i$, let $k = \arg\max_{i \in S} x_i$. Then we have,

   $$\Pr[\text{success}|T^* = t, \mathcal{S} = S] = \Pr[T_k < \tau | T^* = t, \mathcal{S} = S] = \Pr[T_k < \tau | T_k < t] = \tau/t.$$

   Therefore, for any S,

   $$\Pr[\text{success}|T^* = t, \mathcal{S} = S] \geq \tau/t,$$

   which implies that for $t \geq \tau$,

   $$\Pr[\text{success}|T^* = t] \geq \tau/t.$$

   Hence,

   $$\Pr[\text{success}] = \int_0^1 \Pr[\text{success}|T^* = t]dt \geq \int_\tau^1 \frac{\tau}{t} \cdot dt = -\tau \ln \tau.$$

   The lower bound on the success probability is maximized at $\tau = 1/e$, and the maximum value is $1/e$.

2. **[4 points]** Prove that no algorithm for the real-time secretary problem can have success probability $(1/e) + \varepsilon$ for any constant $\varepsilon > 0$.

   Answer: We prove that if there is an algorithm $\mathcal{A}$ for real-time secretary with success probability at least α, then there is an algorithm $\mathcal{A}'$ for secretary with success probability at least α. Since the secretary problem doesn't have an algorithm with success probability $1/e + \varepsilon$ (proved in class), neither does real-time secretary.

   Algorithm $\mathcal{A}'$ knows n. It samples n values from $U[0, 1]$ and sorts them to get $t_1 < t_2 < \cdots < t_n$. For $i = 1$ to n, it applies timestamp t_i to the i'th number in its input and passes it to $\mathcal{A}$. If $\mathcal{A}$ accepts and stops, so does $\mathcal{A}'$.

   Since $\mathcal{A}'$ gets the x_i's in a uniformly random order, arrival times $T_1, \ldots, T_n$ are obtained by sorting n independent draws from $U[0, 1]$ and randomly permuting them again. Thus, $T_1, \ldots, T_n$ are n independent draws from $U[0, 1]$, and hence, the input to $\mathcal{A}$ satisfies the definition of real-time prophet. Therefore, $\mathcal{A}$ succeeds with probability at least α. Algorithm $\mathcal{A}'$ succeeds if and only if algorithm $\mathcal{A}$ succeeds, so $\mathcal{A}'$ succeeds with probability at least α.

Homework 3

In this homework we will analyze a fixed-threshold algorithm for the prophet-secretary problem, for the case when the CDFs $F_1, \ldots, F_n$ of the independent random variables $X_1, \ldots, X_n$ are all continuous, that is, none of the probability distributions have point masses. Observe that this implies that there exists a τ such that $\Pr[\max_i X_i \le \tau] = \prod_{i=1}^n F_i(\tau) = 1/e$. We will analyze the algorithm that uses this τ as the fixed threshold, that is, it accepts the earliest value that exceeds τ. Like in the recorded lectures, we write the algorithm's reward as a sum of revenue and utility.

1. **[1 point]** Determine the expected revenue of the algorithm.

   Answer: The revenue is τ if the algorithm accepts some value, and 0 otherwise. The algorithm accepts some value if and only if at least one X_i is at least τ, that is, $\max_i X_i \ge \tau$. Thus, $\mathbb{E}[\text{rev}] = \tau \cdot \Pr[\max_i X_i \ge \tau] = \tau \cdot (1 - 1/e)$.

2. To analyze the expected utility, it is convenient to imagine that each random variable X_i appears at a uniformly random arrival time t_i in $[0, 1]$, and these n arrival times are independent (like in the real-time prophet-secretary problem defined in the recorded lectures). Let the random variable T denote the stopping time of the algorithm. Like in the prophet secretary analysis, we define $\theta(t) = \Pr[T \ge t]$, the probability that the algorithm doesn't stop before time t.

   (a) **[4 points]** Show that $\theta(t) = \prod_{i=1}^n (1 - t + t \cdot F_i(\tau))$. Hence, prove that $\theta(t) \ge e^{-t}$. (Hint: AM-GM inequality.)

   Answer: The algorithm stops before time t if and only if there exists some i such that $X_i \ge \tau$ and it arrives before time t. In other words,

$$\begin{aligned} \theta(t) = \Pr[T \ge t] &= \Pr[\forall i: \ \neg(t_i \le t \wedge X_i \ge \tau)] \\ &= \prod_{i=1}^n (1 - \Pr[t_i \le t \wedge X_i \ge \tau]) \\ &= \prod_{i=1}^n (1 - \Pr[t_i \le t] \cdot \Pr[X_i \ge \tau]) \\ &= \prod_{i=1}^n (1 - t \cdot (1 - F_i(\tau))) \\ &= \prod_{i=1}^n (1 - t + t \cdot F_i(\tau)), \end{aligned}$$

   where we used the fact that the values X_i and the arrival times t_i are all independent. Next, we have

$$1 - t + t \cdot F_i(\tau) = (1 - t) \cdot 1 + t \cdot F_i(\tau) \ge 1^{1-t} \cdot F_i(\tau)^t = F_i(\tau)^t,$$

   by AM-GM inequality, and hence,

$$\theta(t) \ge \prod_{i=1}^n F_i(\tau)^t = \left(\prod_{i=1}^n F_i(\tau) \right)^t = e^{-t},$$

   by the definition of τ.

   (b) **[4 points]** Observe that for all i, we have $\theta(t) \le \Pr[T \ge t \mid t_i \ge t] = \Pr[T \ge t \mid t_i = t]$. (You don't have to write the proof of this; it is the same as in the recorded lecture.) Using this fact and the bound on $\theta(t)$ you just proved, show that the expected utility is bounded from below by $(1 - 1/e) \cdot \mathbb{E}[(\max_i X_i - \tau)^+]$. (As usual, a^+ denotes $\max(a, 0)$.)

   Answer: The utility is given by

$$\text{util} = \sum_i (X_i - \tau)^+ \cdot \mathbb{I}[\text{ALG picks } X_i] = \sum_i (X_i - \tau)^+ \cdot \mathbb{I}[T \ge t_i],$$

since if $X_i < \tau$ then $(X_i - \tau)^+ \cdot \mathbb{I}[\text{ALG picks } X_i] = (X_i - \tau)^+ \cdot \mathbb{I}[T \geq t_i] = 0$, whereas if $X_i \geq \tau$, then the algorithm picks X_i if and only if $T \geq t_i$. Define $\text{util}_i = (X_i - \tau)^+ \cdot \mathbb{I}[T \geq t_i]$, so that $\text{util} = \sum_{i=1}^n \text{util}_i$. Then we have,

$$
\begin{aligned}
\mathbb{E}[\text{util}_i] &= \mathbb{E}[(X_i - \tau)^+ \cdot \mathbb{I}[T \geq t_i]] \\
&= \mathbb{E}[(X_i - \tau)^+] \cdot \Pr[T \geq t_i] \\
&= \mathbb{E}[(X_i - \tau)^+] \cdot \int_0^1 \Pr[T \geq t \mid t_i = t] dt \\
&\geq \mathbb{E}[(X_i - \tau)^+] \cdot \int_0^1 \theta(t) dt \\
&\geq \mathbb{E}[(X_i - \tau)^+] \cdot \int_0^1 e^{-t} dt \\
&\geq \mathbb{E}[(X_i - \tau)^+] \cdot \left(1 - \frac{1}{e}\right),
\end{aligned}
$$

where the second equality holds because the event $T \geq t_i$ is completely determined by the arrival times and the values of X_j for $j \neq i$, and all these are independent of X_i. Summing over all i, we get,

$$
\begin{aligned}
\mathbb{E}[\text{util}] &= \sum_{i=1}^n \mathbb{E}[\text{util}_i] \\
&\geq \left(1 - \frac{1}{e}\right) \cdot \sum_{i=1}^n \mathbb{E}[(X_i - \tau)^+] \\
&= \left(1 - \frac{1}{e}\right) \cdot \mathbb{E}\left[\sum_{i=1}^n (X_i - \tau)^+\right] \\
&\geq \left(1 - \frac{1}{e}\right) \cdot \mathbb{E}\left[(\max_i X_i - \tau)^+\right],
\end{aligned}
$$

where the second inequality was proved in recorded lecture 7 (see the claim on the second page of notes).

3. **[1 point]** Using the bounds on the expected revenue and the expected utility, derive a competitiveness guarantee of the algorithm.

   Answer: Adding the bounds on expected revenue and expected utility, we get,

$$\mathbb{E}[\text{ALG}] = \mathbb{E}[\text{rev}] + \mathbb{E}[\text{util}] \geq (1 - 1/e) \cdot (\tau + \mathbb{E}[(\max_i X_i - \tau)^+]) = (1 - 1/e) \cdot \mathbb{E}[\tau + (\max_i X_i - \tau)^+],$$

   which implies $\mathbb{E}[\text{ALG}] \geq (1 - 1/e) \cdot \mathbb{E}[\max_i X_i]$, that is, the algorithm is $(1 - 1/e)$-competitive.

Note: the idea can be extended to handle the case when F_i's have point masses. Figure out the details if you are interested.