Today we talk about the secretary problem, online auctions, and finally Ad Auctions.

**Classic secretary problem**: There are $n$ applicants arriving online one by one. After interviewing an applicant $i$, we can decide its value $v_i$ relative to all $1, \ldots, i-1$ (indeed only its relative order is enough). After interviewing applicant $i$, we have to decide to hire $i$ or not: select the best applicant.

One naive algorithm: Randomly choose some element $j$ (give competitive ratio $n$ in the worst case). Choose the first applicant can be decided by adversary, but the order in which applicants arrive is uniformly at random.

Motivations for random permutation model is when the values are coming from an unknown distribution which is practical.

**Algorithm**

- Interview first $k$ applicants and don't hire anyone.
- Let $\max_i$ be the (value of) the best element in first $k$.
- Select the first element after $k$ elements which is better than $\max_i$.

Indeed, above is a stopping rule policy. It is shown by Dynkin 1962 that the optimal policy for the problem is a stopping rule under which we skip the first $k$ applicants and then take the next applicant who is a candidate (i.e., who has the best relative ranking of those interviewed up to that point). For an arbitrary cutoff $k$, the probability that the best applicant is selected is

$$P(k) = \frac{n}{2^n - k} \sum_{j=1}^{n} \frac{(k-1)}{j-1}$$

where $\frac{k}{n}$ is the probability that $j$th element is the maximum and the second highest among places $1$ to $j$ appears in the first $k$ slots.

Letting $n$ tend to infinity as the limit of $k$, using $t$ for $\frac{k}{n}$ and $dt$ for $\frac{1}{n}$, the sum can be approximated by the integral

$$P(x) = \int_{x-1}^{1} \frac{1}{x} \, dt = -\ln x.$$  Take the derivative of $P(x)$ with respect to $x$, setting it to 0 and solving for $x$, we find that the optimal value for $x = \frac{1}{e}$. Thus the optimal cutoff tends to $\frac{1}{e}$ as $n$ increases and the best applicant is selected with probability $\frac{1}{e} = 0.368$.

The problem has been considered for example for unknown number of applicants.
Secretary problem and its extensions have applications in finding a fiancee (the original motivation), hiring secretaries, dynamic auction market, online market, variation of house selling problem, the one-sided bandit problem, detecting a changepoint and Douglas problem, and etc.

A natural extension with applications to online market:
we have to select k secretaries instead of just 1 again in a random permutation model.
different objectives:
1. maximize the probability of selecting best k.
2. minimize the expected sum of ranks of selected k.
3. maximize the expected sum of values of selected k.
we focus mainly on 1 and generalization though 2 and 3 can be modeled as well.

Submodular Secretary Problem
A function f is monotone iff f(A) ≤ f(B) for A ⊆ B ≤ S.
A function F defined on the subsets of a universe set X is submodular if for any pair of subsets of X like A and B, we have

\[ F(A) + F(B) ≥ F(A \cup B) + F(A \cap B) \]

It is subadditive if \[ F(A) + F(B) ≤ F(A \cup B) \]
An equivalent characterization is that the marginal profit of each item should be non-increasing, i.e., \[ f(A \cup \{a\}) - f(A) ≤ f(B \cup \{a\}) - f(B) \] if \( x \in A \setminus B \) and \( \{a\} \subseteq B \)
we want to hire k secretaries such that each subset of secretaries has a value to you which not necessarily linear and this function is submodular in many practical applications.

If F(A) = \[ \mathbb{E} V(a) \] i.e. linearcase, the problem has a 1- \( \Theta(1) \) competitive algorithm. Also the case that \[ F(A) = \max_{a \in A} V(a) \] has been considered before. (Special case)

Thm: There is a constant (\( \Theta(1) \)) competitive algorithm for the (non-monotone) submodular secretary problem. Indeed, this is the most that one can hope for.

For subadditive secretary problem, the ratio can be as bad as \( \Omega(n) \).

The algorithm is simple and intuitive: we partition the input stream into k equal parts: \( S_1, S_2, ..., S_k \) and choose exactly one secretary in each part.
Let \( T_i \) be the set of i secretaries we choose in the first i parts.
In part \( S_i \), we try to choose a secretary \( x \in S_i \) that maximizes \( F(T_i \cup \{x\}) - F(T_i) \) (Using the classic secretary problem
We prove the expected value of \( f(T_k) \) is within a constant factor of the optimum solution. Let \( R = \{ a_{i_1}, a_{i_2}, \ldots, a_{i_k} \} \) be the optimal solution.

Note that the set \( \{i_1, i_2, \ldots, i_k\} \) is a uniformly random subset of \( \{1, 2, \ldots, n\} \) with size \( k \) (due to random ordering).

First by submodularity, we have \( f(D) - f(A) \leq \sum f(A_{i_j} f(A) \text{ for any } A \subseteq B \subseteq S \) \( \forall A \in D \).

Define \( X = \{ s_i : s_i \cap R \neq \emptyset \} \). For each \( s_i \in X \), we pick one element \( s_i \in s_i \) of \( s_i \cap R \) random by \( \log_2 \). These selected items form a set called \( R' = \{ s_1, s_2, \ldots, s_{|X|} \} \) \( \subseteq B \) of size \( k \).

Lemma 1: The expected value of the number of items in \( R' \) is at least \( k(1 - \frac{1}{e}) \).

Proof: By standard methods (like balls and bins), we can show at most \( \frac{1}{e} \) fractions of \( s_i \) have empty intersection with \( R \).

Another intuitive Lemma 2: For a random subset \( A \) of \( R \), the expected value of \( f(A) \) is at least \( \frac{1}{|A|} f(R) \).

By randomness and submodularity, the crux of the proof is that local optimizations steps indeed lead to global approximate solution, i.e.,

Lemma 3: The expected value of \( f(T_k) \) is at least \( \frac{|R'|}{|R|} f(R) \).

Intuition of the proof: Assume w.l.o.g. \( s_1 \) is the first element of \( R \) to appear in our ordering. Define \( \Delta_j = f(T_j) - f(T_{j-1}) \) note that due to monotonicity \( \Delta_j \geq 0 \) and thus \( E[\Delta_j] \geq 0 \) with probability \( \frac{1}{e} \), we choose the element of \( s_1 \) which maximizes the value of \( f(T_j) \) (given that the set \( T_{j-1} \) is fixed). We get factor \( \frac{|R'|}{|R|} \) \( f(R) \).

Since by (Lemma 2) we can relate the value of \( f(T_k) \) with \( O(f(R)) \) which is by Lemma 2 is \( \frac{|R'|}{|R|} f(R) \).

Finally, the expected value of the output of our algorithm is at least \( \frac{|R'|}{k} f(R) \).

Proof: Use Lemma 1 that expected value of \( R' \) is \( k(1 - \frac{1}{e}) \). The exact number is \( \frac{|R'|}{k} f(R) \).
For non-monotone case, this algorithm does not work, since if we select some item for the ith set, it may hurt us at a later step. However, slight changes works.

Let divide the input stream into two equal parts and let $0 < x < 1$ be a uniformly random value. If $x \leq \frac{1}{2}$, we run the algorithm on the first part (and thus ignore the other part), otherwise we run the algorithm on the second part and ignore the first part. One can prove that the sum of the solutions in two parts is a constant times the opt and thus we get at least half of it.

Next we consider the applications in online auctions of secretary problem.