

Hiring Problem

- An algorithm is *randomized* if its behavior is determined not only by its input but also by values produced by a *random-number generator*.
- $\text{RANDOM}(a, b)$
 - returns an integer between a and b , inclusive with each such integer being equally likely.
 - independent of the integers returned on previous calls
- For e.g. $\text{RANDOM}(0, 1)$ generates 0 or 1 with probability $1/2$
 $\text{RANDOM}(3, 7)$ returns either 3, 4, 5, 6 or 7, each with probability $1/5$.

Need an office assistant

- Employment Agency sends one candidate every day
- Interview that person, either hire that person (and fire the old one), or keep old person
- Always want the best person – always hire if interviewee is better than current person

- Number of candidates = n .
- Interview one candidate per day decide whether to hire the person or not (Cost of interviewing a candidate = α)
- If the candidate is better than the current assistant, must fire the assistant and hire the candidate. (Cost of hiring any candidate = ch .)
- **Goal: Estimate the cost of this strategy**

Assumption

- The candidates for the office assistant job are numbered 1 through n .
- creates a dummy candidate, numbered 0, who is less qualified than each of the other candidates.
- after interviewing candidate i , determine if candidate i is the best candidate you have seen so far.

HIRE-ASSISTANT(n)

```
1   $best \leftarrow 0$       ▷ candidate 0 is a least-qualified dummy candidate
2  for  $i \leftarrow 1$  to  $n$ 
3      do interview candidate  $i$ 
4          if candidate  $i$  is better than candidate  $best$ 
5              then  $best \leftarrow i$ 
6              hire candidate  $i$ 
```

Cost associated to this problem:

- Since all candidates must be interviewed, the interview cost $n\alpha$ is unavoidable.
- Hiring cost varies with the number of candidates hired. So, we focus on this cost. If 'm' number of candidates are hired, then hiring cost is mc_h .
- **Total cost = $O(n\alpha + mc_h)$**
- Interview cost – Less – E.g., payment towards employment agency
- Hiring cost – High – E.g., cost of firing old employee (salary and other payment settlements), Training of hired employee
- In hiring problem, certain ID is given to the candidates.
- We assume that candidates appear in a random order.
- So, we can have $n!$ permutations. If $n = 1$, only one possibility. If $n = 2$, then we have 2 possibilities i.e., (1,2) or (2,1). If $n = 3$, then we have 6 possibilities i.e., (1,2,3), (1,3,2), (2,1,3), (2,3,1), (3,1,2), (3,2,1) and likewise for all values of n .
- Each permutation has equal probability to appear as a sequence for the interview.
- But what if they are not appearing in random order. E.g., employment agency may get some charges from the candidates if they are hired and may fool the employer by sending the candidates in an order such that

candidate sent every day appear best when compared with other previously hired.

- Worst case scenario occurs.

Worst-case analysis:

- Candidate list is in increasing order of quality; that is, candidate i is better than candidate $i - 1$, $1 \leq i \leq n$.
- Every candidate is hired. Therefore, hiring cost $= nc_h$.

- To avoid such situations, we can use randomized algorithm to bring out more control over the input sequence.
- We force all inputs to be equally likely, by randomizing the input.
- In the hiring problem, we randomly permute the lists of candidates and then run the algorithm.

Randomized algorithm for Hiring Assistant:

Assumptions:

- The list of candidates is an input to the algorithm.
- Algorithm randomly permutes the list to generate the order in which candidates are interviewed.

RANDOMIZED-HIRE-ASSISTANT(n)

```
1  randomly permute the list of candidates
2   $best \leftarrow 0$       ▷ candidate 0 is a least-qualified dummy candidate
3  for  $i \leftarrow 1$  to  $n$ 
4      do interview candidate  $i$ 
5          if candidate  $i$  is better than candidate  $best$ 
6              then  $best \leftarrow i$ 
7              hire candidate  $i$ 
```

The expected hiring cost of the procedure RANDOMIZED-HIRE-ASSISTANT is $O(c_h \ln n)$.

Randomized Hiring Assistant Problem-Complexity Analysis

- Use indicator random variables; let x_i denote the indicator variable for candidate i . (So, $x_i = 1$ if the candidate is hired; $x_i = 0$ otherwise.)
- If X denotes the random variable that represents the number of candidates hired, then $X = \sum_{i=1}^n x_i$. We want to compute $E[X]$.

$E[X]$ -> expected number of hirings

$E[x_i]$ -> Probability that the i th candidate is best of first i

- By linearity of expectation, $E[X] = \sum_{i=1}^n E[x_i]$.
- $\Pr\{x_i = 1\} = 1/i$. So, $E[x_i] = 1/i$, $1 \leq i \leq n$.

- What is $E[X_i]$?
 - $E[X_i]$ = Probability that the i th candidate is hired
 - i th candidate hired when s/he is better than the $i - 1$ candidates that came before
 - Assuming that all permutations of candidates are equally likely, what is the probability that the i th candidate is the best of the first i candidates?
 - $1/i$

- Probability that the i th candidate is best of first i is $1/i$
 - Sanity Check: (Doing a few concrete examples as a sanity check is often a good idea)
 - $i = 1$, probability that the first candidate is the best so far = $1/1 = 1$
 - $i = 2$: (1,2), (2,1) In one of the two permutations, 2nd candidate is the best so far
 - $i = 3$: (1, 2, 3), (1, 3, 2), (2, 1, 3), (2, 3, 1), (3, 1, 2), (3, 2, 1) In two of the 6 permutations, the 3rd candidate is the best so far

$$\begin{aligned}
&= \sum_{i=1}^n E[x_i] \\
&= \sum_{i=1}^n 1/i \\
&= \ln n + O(1) \quad \text{-- For Positive Integers, the } n\text{th Harmonic series} \\
&\in O(\lg n)
\end{aligned}$$

Therefore, $T(n) = O(\log n)$