VALLURUPALLI NAGESWARA RAO VIGNANA JYOTHI INSTITUTE OF ENGINEERING AND TECHNOLOGY

BACHUPALLY (Via), KUKATPALLY – 500090

TAMASOMA JYOTIRGAMAYA

NAME: G. SAHITH REDDY

ROLL NO.: 21071A6718

DEPARTMENT OF CSE-DS, CYS AND AI&DS

FLAT ASSIGNMENT

Hiring Problem:

For example, you are using an employment agency to hire a new assistant for your company. The agency works in such a way that it sends you one candidate per day. You then interview the candidate after which you must immediately decide whether or not to hire that person. Also, if you hire that person, you must fire your current office assistant. Even if it is someone you have hired recently you must fire them before you hire a new assistant.

let us consider the cost to interview as ci per candidate and the cost to hire the candidate is ch.

Also, there are a few requirements to be fulfilled which are:

At all times, you want to have the best candidate you have seen so far.

Whenever you interview a candidate and feel that the candidate is better than your current assistant, you have to fire the current assistant and then hire the candidate.

You should also always hire the first candidate that you interview.

Pseudo code to the model.

Hire-Assistant (n)

best <-0 ;Candidate 0 is a least qualified candidate

for i <- 1 to n

do interview candidate i

if candidate i is better than candidate best

then best <- i

hire candidate i

Here are a few points to be considered for the Cost Model:

We are not concerned about the running time of the Hire-Assistant

We must determine the total cost of hiring the best candidate.

If n candidates are interviewed and m candidates are hired, then the cost would be nci+mce

We have to pay the cost: nci to interview the candidates' irrespective oof how many candidates we hire.

So, focus on analyzing the hiring cost mch

m varies with the order of the candidates.

Best-Case Analysis for Hiring Problem

We get the best case when the agency sends us the best applicant on the first day.

The cost would be nci+c

Worst-case Analysis for Hiring problem

The worst case is when we hire all the n candidates.

This happens only if each candidate that comes later is better than all those who came before. That is when candidates come in increasing order of quality.

The cost would be nc_i+nc_h

Whenever this happens, we will fire the agency.

Average-case Analysis for Hiring problem

For the average case, an input to the hiring problem is an ordering of the n applicants, there are n! different inputs.

To use probabilistic analysis, we assume that the candidates arrive in random order.

Then we analyze our algorithm by computing an expected running time.

This expectation is taken over the distribution of all the possible inputs.

Thus, we are averaging the running time over all possible inputs.

Now, we want to know the expected cost of our hiring algorithm in terms of the number of times we hired an applicant.

For this, Random variable X(s) is the number of applicants who are hired for a given input sequence s.

Indicator random variable X_i for applicant i will be 1 if applicant i is hired, 0 otherwise.

What is Indicator Random Variable?

It is a powerful technique that is used for computing the expected value of a random variable. Also, it is a convenient method for converting between probabilities and also expectations. Indicator Random Variable takes only 2 values, which are 1 and 0.

Indicator Random Variable $X_A=I\{A\}$ for an event A of a sample space is defined as:

$I\{A\}=$ 1 if A occurs

0 if A does not occur

The expected value of an indicator Random Variable is associated with an event A is equal to the probability that A occurs.

Indicator RV – Example

Problem: Determine the expected number of heads in one toss.

Sample space is s{H, T}

Indicator random variable

XA = I{coin coming up with heads}=1/2

The expected number of heads obtained in one flip of the coin is equal to the expected value of the indicator random variable.

Therefore, E[XA] = 1/2

```
best ← 0    ®candidate 0 is a least-qualified dummy candidate
ori ← 1 to n
    do interview candidate i
      if candidate i is better than candidate best
        then best ← i
             hire candidate i
```

```
import math
import random
def random_number(a, b):
   bits = math.ceil(math.log2(b - a + 1))
   while True:
      number = random_binay(bits)
```

```
        if a + number <= b:
            return a + number
def random_binay(bits):
    number = 0
    for i in range(bits):
        number = number * 2 + random.randint(0, 1)
    return number
```