Experiment 2

AA Experiment DATE:

Kreena Shah
60004210243
C'32

Aim : To implement hiring problem (randomized algorithms)

Theory :

No. of candidates = n
Cost of interviewing a candidate = C_h
If the candidate is better than the current assistant, must fire the assistant & hire the candidate

To estimate cost :
Since all cost candidates must be interviewed, the interview cost n is unavoidable
If m candidates are hired, the hiring cost is much. This cost varies with the input

```
HireAssistant(n)
    best ← 0
    for i ← 1 to n
        do interview candidate i
            if candidate i is better than candidate best
                then best ← i
                    hire candidate i
```

Worst Case Analysis :
- Candidate list is in increasing order of quality i.e. candidate i is better than candidate i-1 $(1 \leq i \leq n)$

- Every candidate is hired
  ∴ Hiring cost = nC_h

To avoid this, we randomize the list of candidates

Code & Output :

```python
import random

candidates = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
interviewed_candidates = []
hired_candidates = []

# Randomly select and interview candidates
for i in range(len(candidates)):
    selected_candidate = random.choice(candidates)
    interviewed_candidates.append(selected_candidate)
    candidates.remove(selected_candidate)

# Hire the best candidate so far
max=-1
for i in range(len(interviewed_candidates)):
    if interviewed_candidates[i] > max:
        max=interviewed_candidates[i]
        hired_candidates.append(interviewed_candidates[i])

# Calculate firing cost
firing_cost = len(hired_candidates) - 1

print("Interviewed candidates:", interviewed_candidates)
print("Hired candidates:", hired_candidates)
print("Number of candidates hired:", len(hired_candidates))
print("Firing cost:", firing_cost)
```

```
PS C:\Users\Admin\OneDrive\Desktop\sem6\AA\Pracs> py .\hiring.py
Interviewed candidates: [0, 2, 5, 9, 7, 8, 4, 3, 6, 1]
Hired candidates: [0, 2, 5, 9]
Number of candidates hired: 4
Firing cost: 3
PS C:\Users\Admin\OneDrive\Desktop\sem6\AA\Pracs>
```

Conclusion : Thus, we implemented hiring problem using a randomized algorithm