Data Analysis and Algorithm

Practical 4

Implement Hiring Problem and analyze its

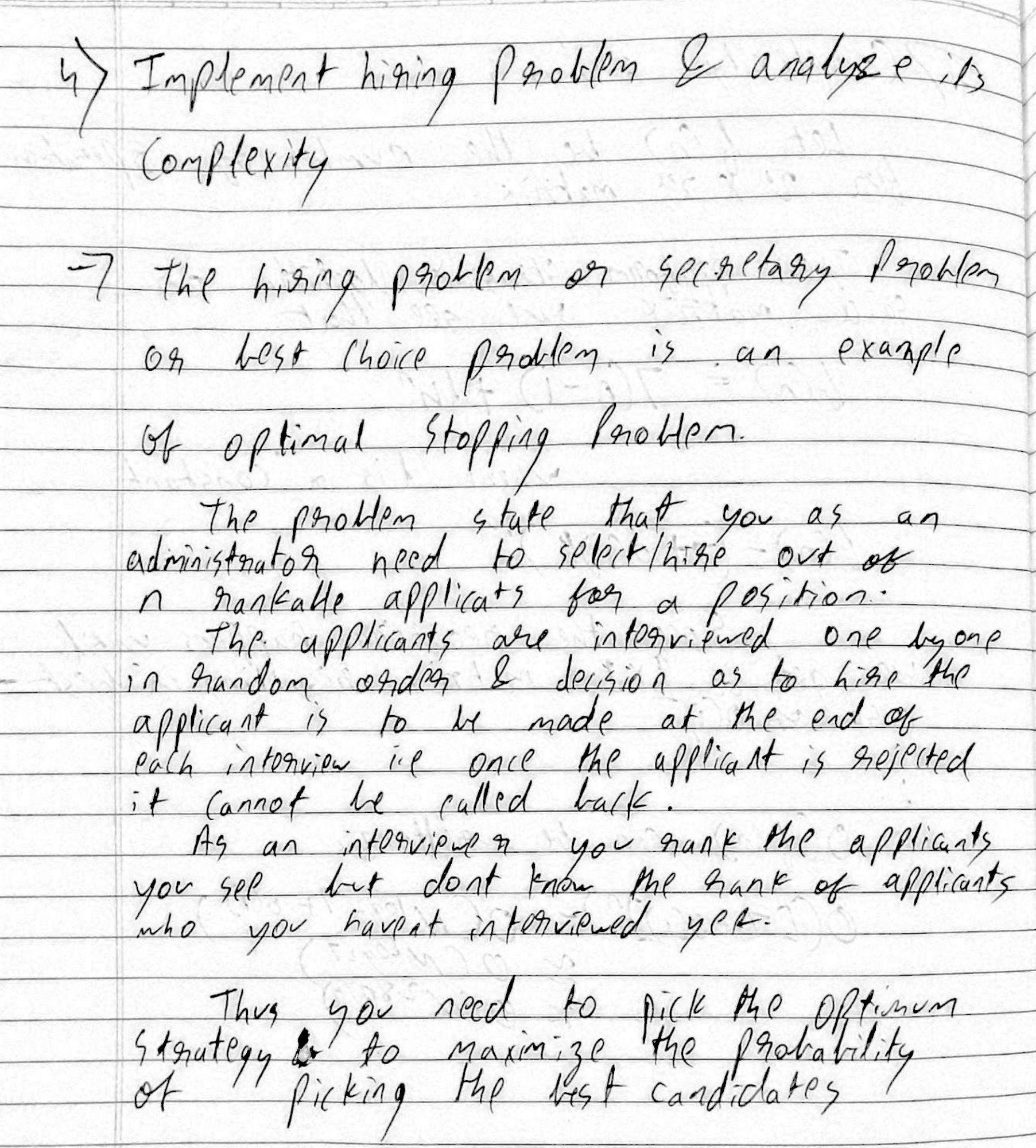
complexity.

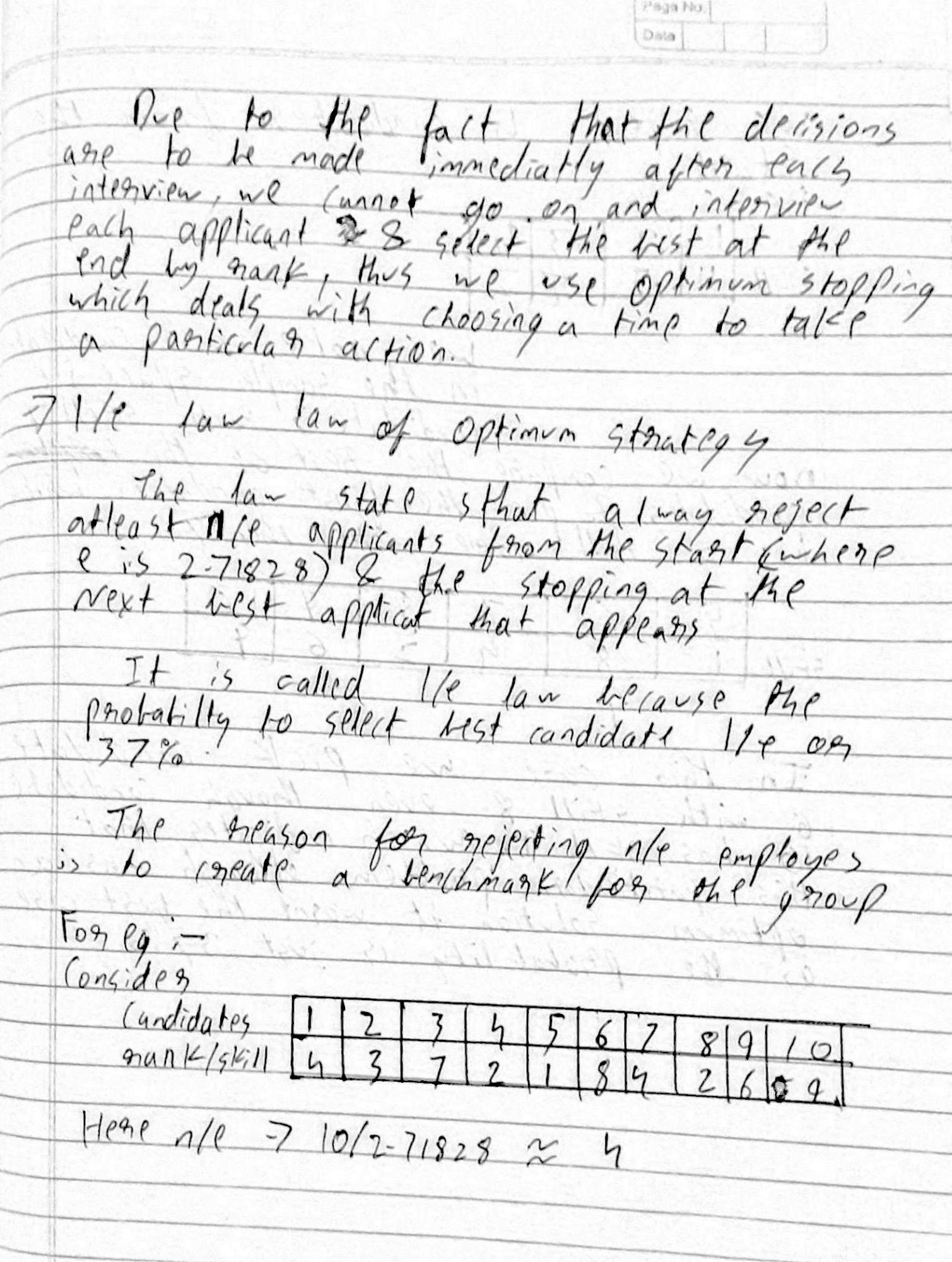
Date.: 16-09-21

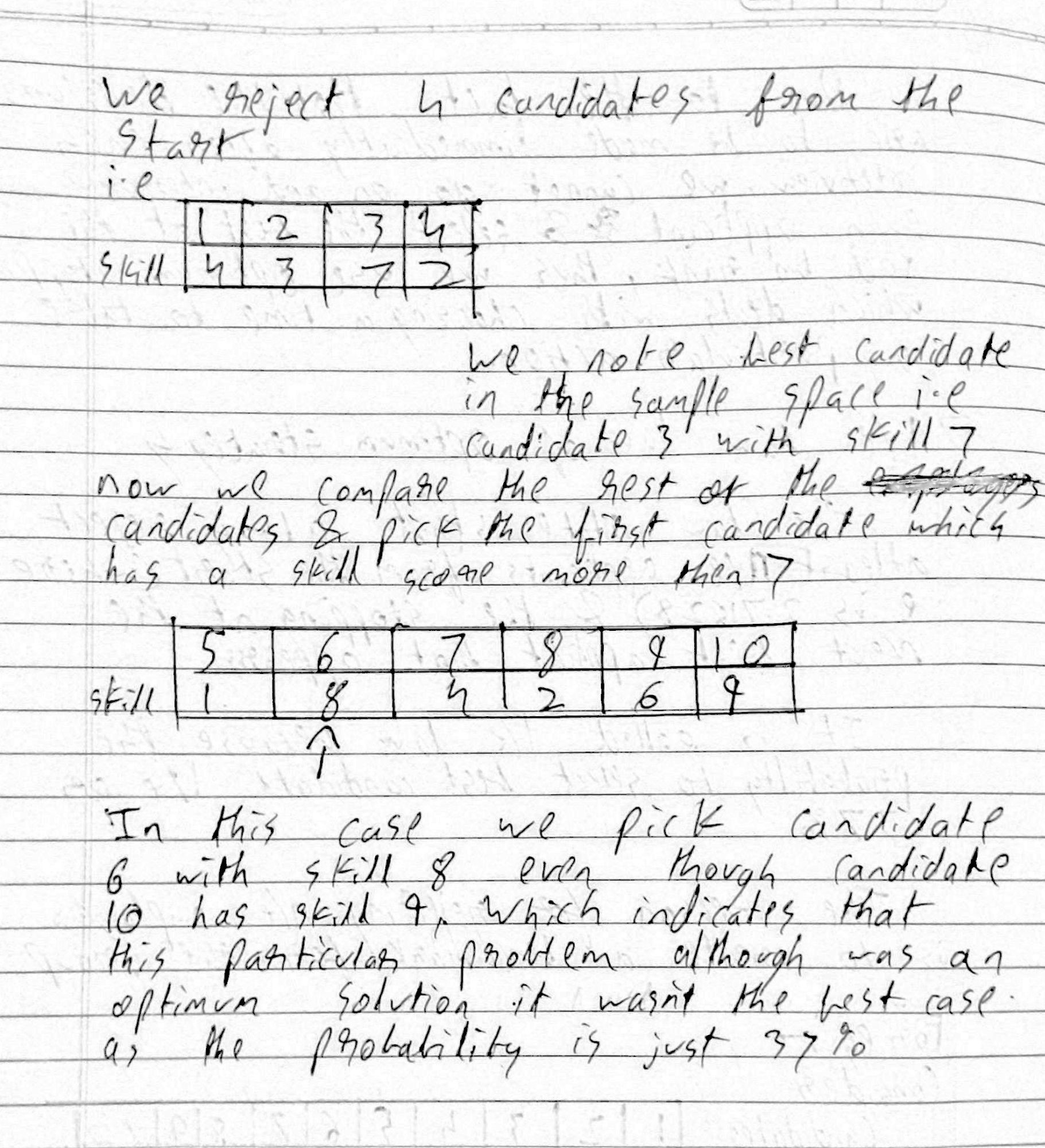
Name – Yash Vasudeo Prajapati

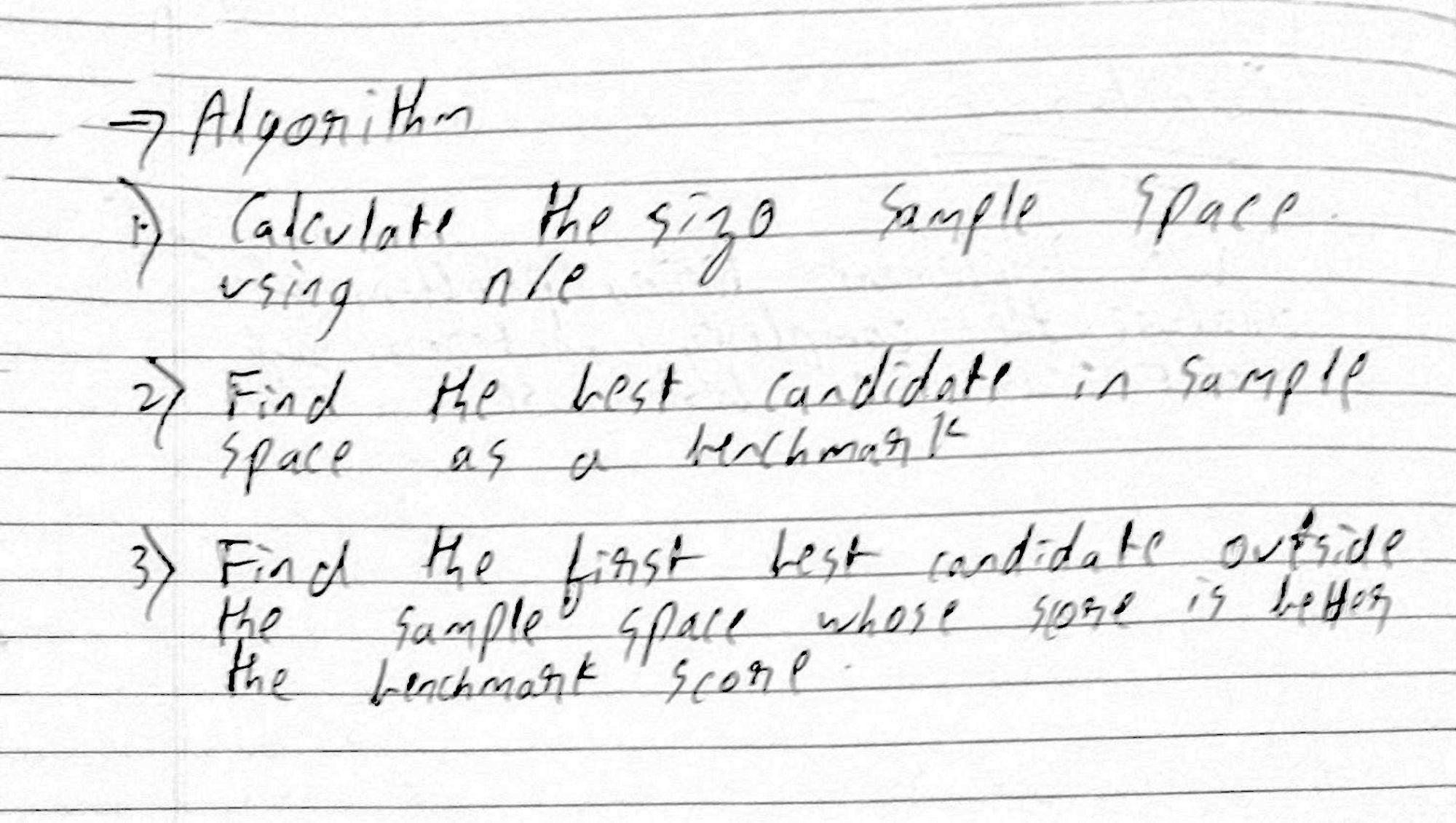
Rollno - 022

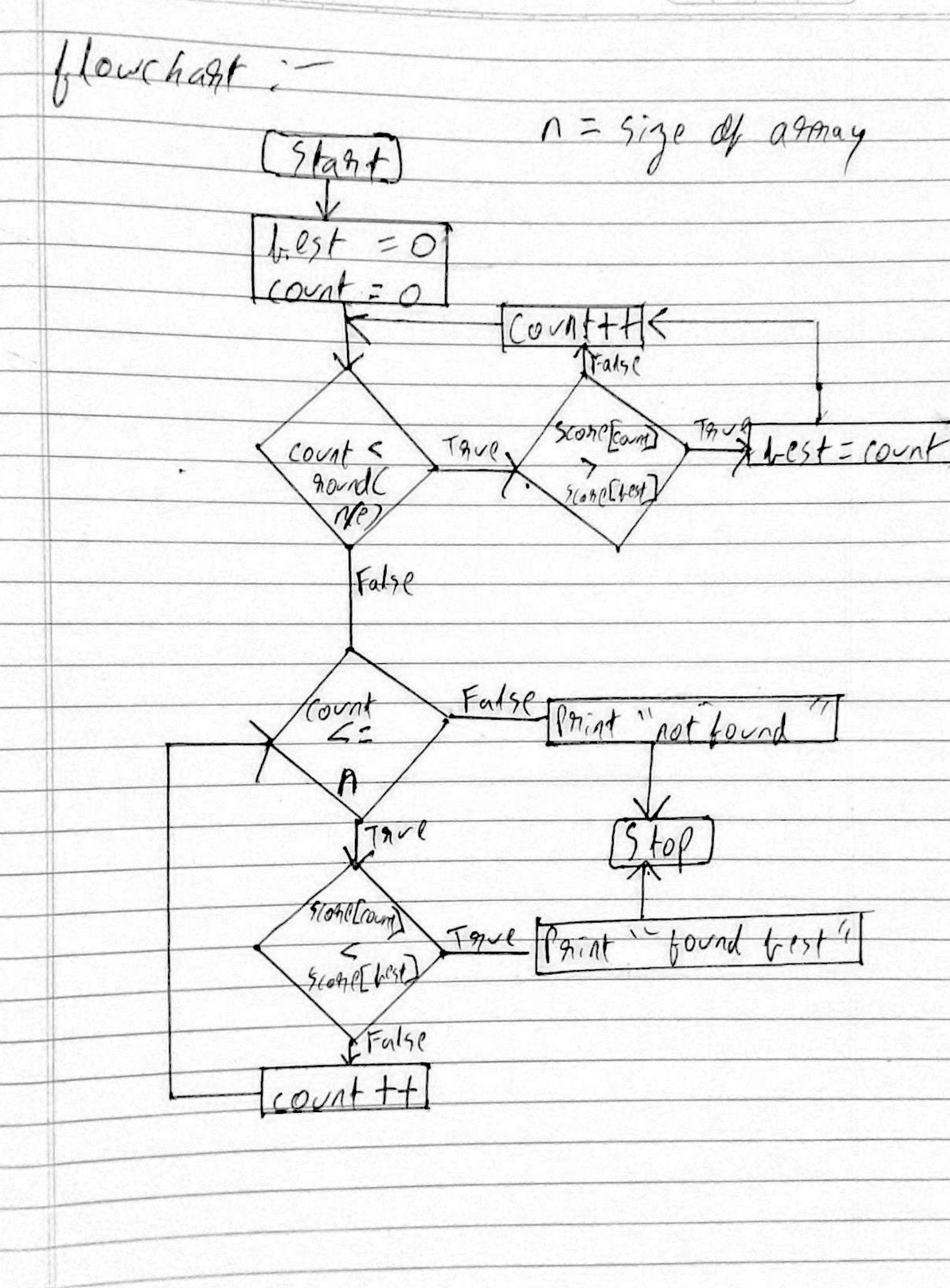
MSc. Computer Science

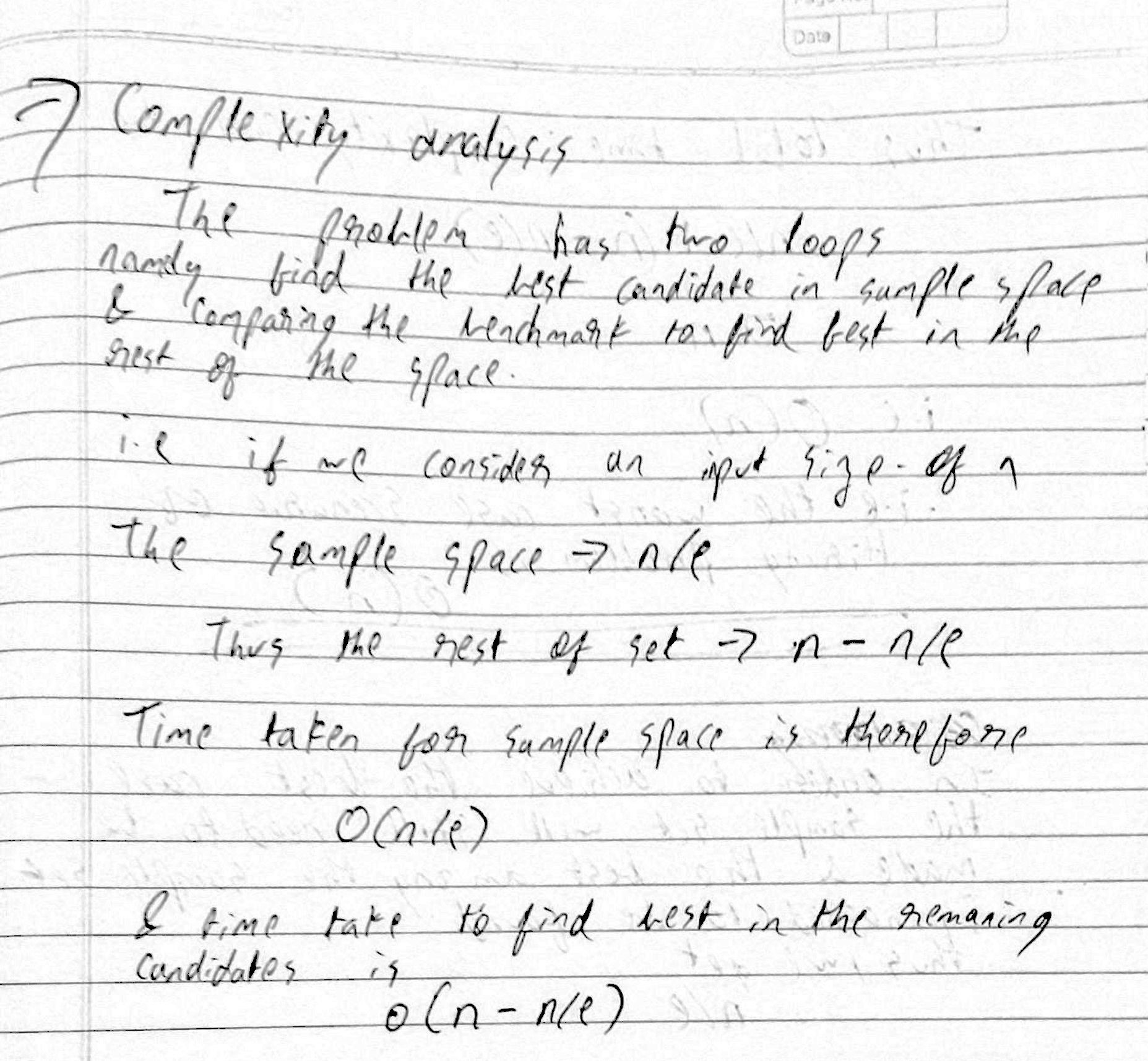


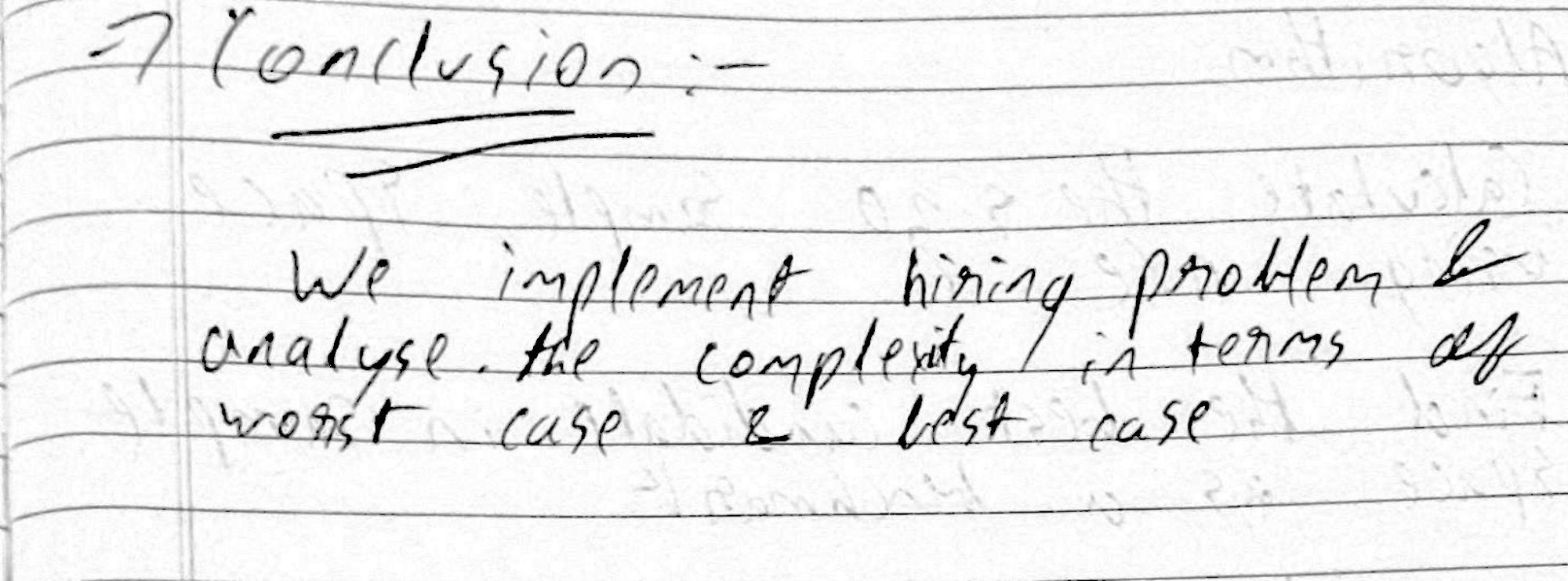
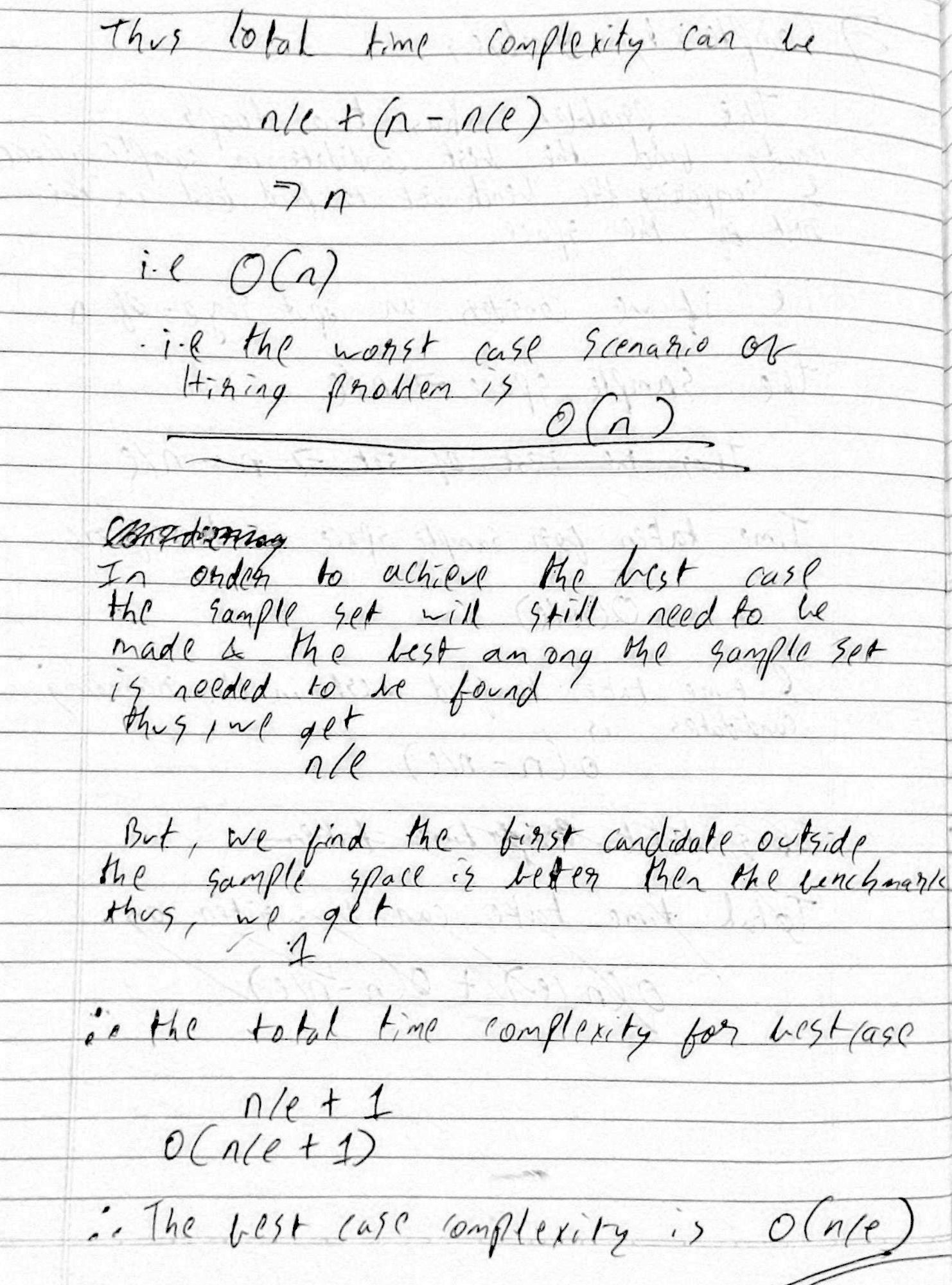








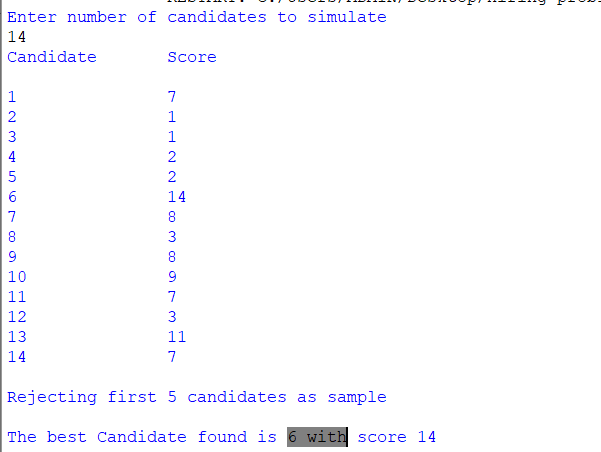




**Program :-**

1. import random
3. def HiringProblem(score, n):
4. sample\_size = int(round(n / e))
5. print(f"\nRejecting first {sample\_size} candidates as sample")
7. #finding best candidate in the sample set for benchmark
8. best\_candidate = 0;
9. for i in range(1, sample\_size):
10. if (score[i] > score[best\_candidate]):
11. best\_candidate = i
13. #finding the first best candidate outside the sample set
14. for i in range(sample\_size, n):
15. if (score[i] >= score[best\_candidate]):
16. best\_candidate = i
17. break
19. if (best\_candidate >= int(sample\_size)):
20. print(f"\nThe best Candidate found is {best\_candidate+1} with score {score[best\_candidate]}")
21. else:
22. print("Couldn't find a best candidate")
24. if \_\_name\_\_ == "\_\_main\_\_":
25. e = 2.71828
26. n = int(input("Enter number of candidates to simulate\n"))
27. score = []
28. #populating the list
29. for i in range(n):
30. score.append(random.randint(1, n))
31. print("Candidate\tScore\n");
33. for i in range(n):
34. print(f"{i+1}\t\t{score[i]}");
35. HiringProblem(score, n);

No of candidates = 14



No of candidates = 100

