Republic of the Philippines
CAMARINES SUR POLYTECHNIC COLLEGES
Nabua, Camarines Sur

COLLEGE *of* COMPUTER STUDIES

CCCS 104 - Data Structures and Algorithms

LEARNING TASK (SEARCHING)

GROUP NO: 9 SECTION: BSCS2A

GROUP LEADER: Venn P. Delos Santos

GROUP MEMBERS: Marc Christian D. Tumaneng

John Mark A. Pajenago

RATIONALE

Searching is a method or technique that assists in locating the location of a particular element or value from any data structure it is stored. It refers to the process of checking and retrieving the needed information from a group of elements which can be in various forms, such as an array, tree, graph, or linked list. A search is considered to be successful or unsuccessful based on whether or not the element being searched is located. Some of the standard searching techniques that are being implemented in data structures are the Linear search and the Binary search. The former is the simplest method for searching. In this technique of searching, the element to be found in searching the elements to be found is searched sequentially in the list. This method can be performed on a sorted or an unsorted list which is usually in arrays. On the other hand, Binary search is used for searching an item in a sorted array. It operates with the same principles as the Divide and Conquer algorithm where it looks for a particular element by comparing the middle most element of the data structure where it is stored.

The Python programs that we made implement Search Algorithms such as Linear Search and Binary Search in order to solve the Thieves & Policemen and Road to Playoffs problems. The Python program, which implements Linear Search in particular, solves and finds the maximum total number of thieves that can be caught by the policemen based on the given input of the variable K units of distance which is important in determining the maximum reach of the policemen. On the other Python program, we used Binary Search in order to get the total number of teams which had a probability greater than zero in qualifying to the playoffs based on given inputs.

SEARCH PROBLEMS and SOLUTIONS

PROBLEM 1: LINEAR SEARCH (Policemen and thieves)

PROGRAM CODE

```
# BSCS2A - Group 9
# Python program to get the maximum number of thieves that can be caught

# Define function and set parameters
def policemen_thieves(K, list1):

    # Counter
    counter = 0

    # Search and count the maximum number of thieves that can be caught using
for loop
    for i in range(len(list1)):
        if list1[i] == 'T' and 'P' in list1:
            if i-K >= 0:
                front = list1[i-K:i]
            else:
                front = list1[0:i]
            if i+K+1 <= len(list1):
                end = list1[i+1:i+K+1]
            else:
                end = list1[i+1:len(list1)]
            if 'P' in front:
                counter = counter + 1
                if i-K >= 0:
                    front_index = (i-K)+front.index('P')
                else:
                    front_index = front.index('P')
                list1[front_index] = 0
            elif 'P' in end:
                counter = counter + 1
                end_index = i+1+end.index('P')
                list1[end_index] = 0

    return counter

# Get user input for the number of test cases
test_case = int(input("Enter the Number of Test Cases: \n"))
for _ in range(test_case):
    # Get the number of N lines and the K units away based from the user input
```

```
N, K = map(int, input().split())
total = 0
for _ in range(N):
    list1 = [ x for x in input().split() ]
    total = total + policemen_thieves(K, list1)

# Display the total of the maximum number of thieves that can be caught
print("\nMaximum thieves that can be caught: ", total)
print("")
```

TEST CASES

INPUT	OUTPUT
5 3 1 P T P T P T T T P	3
4 2 P T P T P P T T P T T T T P P P	6
5 1 P T P T P P P T P P T T P P T P P P T T T P T P T	8

5 3 P T P T P P P T P P T T P P T P P P T T T P T P T 6 4 T T P P T T T T T P P P P P T T T T T P T P T P P P T P T T P T T P T T	9 15

Explanation (Test Case 1)

Total Thieves = 5
Number of thieves reachable by policemen in Row 1 = 1
Number of thieves reachable by policemen in Row 2 = 2
Number of thieves reachable by policemen in Row 3 = 1
However, one policeman can catch at most 1 thief. Hence, in Row 2, only 1 thief is catchable. **Therefore, the 3 thieves can be caught.**

Explanation (Test Case 2)

Total Thieves = 6
Number of thieves reachable by policemen in Row 1 = 2
Number of thieves reachable by policemen in Row 2 = 2
Number of thieves reachable by policemen in Row 3 = 3
Number of thieves reachable by policemen in Row 4 = 1
However, one policeman can catch at most 1 thief. Hence, in Row 1, 2 thieves are catchable. In Row 2, 2 thieves are catchable. In Row 3, only 1 thief is catchable. And in Row 4, only 1 thief is catchable. **Therefore, the 6 thieves can be caught.**

Explanation (Test Case 3)

Total Thieves = 11
Number of thieves reachable by policemen in Row 1 = 2
Number of thieves reachable by policemen in Row 2 = 1
Number of thieves reachable by policemen in Row 3 = 3
Number of thieves reachable by policemen in Row 4 = 2
Number of thieves reachable by policemen in Row 5 = 3
However, one policeman can catch at most 1 thief. Hence, in Row 1, 2 thieves are catchable. In Row 2, only 1 thief is catchable. In Row 3,2 thieves are catchable. In Row 4,2 thieves are catchable. And in Row 5,2 thieves are catchable.**Therefore, the 8 thieves can be caught**.

Explanation (Test Case 4)

Total Thieves = 11
Number of thieves reachable by policemen in Row 1 = 2
Number of thieves reachable by policemen in Row 2 = 1
Number of thieves reachable by policemen in Row 3 = 3

Number of thieves reachable by policemen in Row 4 = 2 Number of thieves reachable by policemen in Row 5 = 3 However, one policeman can catch at most 1 thief. Hence, in Row 1, 2 thieves are catchable. In Row 2, only 1 thief is catchable. In Row 3,2 thieves are catchable. In Row 4,2 thieves are catchable. And in Row 5,2 thieves are catchable.**Therefore, the 9 thieves can be caught.**
Explanation (Test Case 5)
Total Thieves = 21 Number of thieves reachable by policemen in Row 1 = 4 Number of thieves reachable by policemen in Row 2 = 3 Number of thieves reachable by policemen in Row 3 = 4 Number of thieves reachable by policemen in Row 4 = 3 Number of thieves reachable by policemen in Row 5 = 3 Number of thieves reachable by policemen in Row 6 = 4 However, one policeman can catch at most 1 thief. Hence, in Row 1, 2 thieves are catchable. In Row 2, 3 thieves are catchable. In Row 3,2 thieves are catchable. In Row 4,3 thieves are catchable. In Row 5,3 thieves are catchable.And in Row 6,2 thieves are catchable. **Therefore, the 15 thieves can be caught.**

PROBLEM 2: BINARY SEARCH (Road to playoffs)

PROGRAM CODE

```
# BSCS2A - Group 9
# Python program to get the total number of qualified teams for the playoffs

import math

# Define playoffs as function
def playoffs():
    # Get the number of test cases from the user
    test_case = int(input("Enter the Number of Test Cases: \n"))

    # While loop to run how many test cases were given
    while test_case > 0:
        test_case -= 1

        # Take the inputs for the N, M, K, B separated by space
        N, M, K, B = map(int, input().split())
        list1 = list(map(int, input().split()))

        # Sort the list for it to be in order which is a must in Binary search
        list1.sort(reverse = True)

        LB = B-1
        UB = N
        while UB - LB > 1:
            mid = (UB + LB) // 2
            a = M * (K - (B-1) - (N-1))
            for i in range(B-1, mid):
                if list1[i] > list1[mid] + M:
                    a = math.inf
                else:
                    a -= list1[mid] + M - list1[i]
                if a > 0:
                    UB = mid
                else:
                    LB = mid

        # Display the number of qualified teams for the playoffs
        print("\nTeams that can make it into the playoffs: ", LB + 1)
        print("")
```

```
# function call
playoffs()
```

TEST CASES

INPUT	OUTPUT
5 4 2 2 1 4 1 2 3 4 1 1 1 4 4 4 4 6 2 2 1 6 5 4 1 4 4 6 3 3 1 3 4 6 5 3 4 8 1 1 1 5 5 5 5 5 5 5 5	3 4 5 6 8

Explanation (Test Case 1)

Only one team can make it to the playoffs, two days of league stage are left and 2 points on each day are to be gained. Every team has the probability to qualify for playoffs other than a team with 1 point. Let us look at the scenarios:
Bold are those teams which we want to win:
Team with 4 points: [**4**,**1**,2,3] -> [**5**,**2**,2,3] -> [5,3,2,3]
Team with 1 point has no chance to qualify (Zero probability)
Team with 2 points: [4,**1**,**2**,3] -> [4,**2**,**3**,3] -> [4,3,4,3] (Any of the first or third team can qualify so probability>0)
Team with 3 points: [4,**1**,2,**3**] -> [4,**2**,2,**4**] -> [4,3,2,5] 4th team has maximum points so they can be chosen in this case.
So, all teams other than the second have a non-zero chance, and also note that in case of ties you can qualify any team you want.

Explanation (Test Case 2)

Only one team can make it to the playoffs, there is 1 day left of the league stage and 1 point on each day is to be gained. Every team has the probability to qualify for playoffs as their scores are all tied and no team has a score of 1 point.
As all teams have the same scores, all of their scenarios will look like this:
ALL TEAMS: [**4**, **4**, **4** ,**4**] -> [**5**, **5**, **5**, **5**].
So, all 4 teams can qualify as their scores are all tied and there's no team with 1 point.

Explanation (Test Case 3)
Only one team can make it to the playoffs, two days of league stage are left and 2 points on each day are to be gained. Every team has the probability to qualify for playoffs other than a team with 1 point. Let us look at the scenarios: Bold are those teams which we want to win: Team with 6 points: [**6**,**5**,4,1,4,4] -> [**7**,**6**,4,1,4,4] -> [**8**,**7**,4,1,4,4] 1st team has maximum points so they can be chosen in this case. Team with 5 points: [6,**5**,**4**,1,4,4] -> [6,**6**,**5**,1,4,4] -> [6,**6**,**6**,1,4,4] (Any of the first, second, or third team can qualify so probability>0) Team with 4 points: [6,5,**4**,**1**,4,4] -> [6,6,**5**,**2**,4,4] -> [6,5,**6**,**2**,4,4] (Any of the first or third team can qualify so probability>0) Team with 1 point has no chance to qualify (Zero probability) Team with 2 points: [6,5,4,1,**4**,**4**] -> [6,5,4,1,**5**,**5**] -> [6,5,4,1,**6**,**6**] (Any of the first, fifth, or sixth team can qualify so probability>0) Team with 3 points: [6,5,**4**,1,4,**4**] -> [6,5,**5**,1,4,**5**] -> [6,5,**6**,1,4,**6**]] (Any of the first, third or sixth team can qualify so probability>0) So, all teams other than the fourth team have a non-zero chance, and also note that in case of ties you can qualify any team you want.
Explanation (Test Case 4)
Only one team can make it to the playoffs, three days of the league stage are left and 3 points on each day are to be gained. Every team has the probability to qualify for playoffs other than a team with 1 point. Let us look at the scenarios: Bold are those teams which we want to win: Team with 3 points: [**3**,**4**,6,5,3,4] -> [**4**,**5**,6,5,3,4] -> [**5**,**6**,6,5,3,4] -> [**6**,**7**,6,5,3,4] 2nd team has maximum points so they can chosen in this case. Team with 4 points: [3,**4**,**6**,5,3,4] -> [3,**5**,**7**,5,3,4] -> [3,**6**,**8**,5,3,4] -> [3,**7**,**9**,5,3,4] Any of the 2nd and 3rd team can qualify so probability > 0. Team with 6 points: [3,4,**6**,**5**,3,4] -> [3,4,**7**,**6**,3,4] -> [3,4,**8**,**7**,3,4] -> [3,4,**9**,**8**,3,4] 3rd team has maximum points so they can be chosen in this case Team with 5 points: [3,4,6,**5**,**3**,4] -> [3,4,6,**6**,**4**,4] -> [3,4,6,**7**,**5**,4] -> [3,4,6,**8**,**6**,4] 4th team has maximum points so they can be chosen in this case Team with 3 points: [**3**,4,6,5,**3**,4] -> [**4**,4,6,5,**4**,4] -> [**5**,4,6,5,**5**,4] -> [**6**,4,6,5,**6**,4] Any of the 1st or 5th team can qualify so probability>0. Team with 4 points: [3,4,6,5,**3**,**4**] -> [3,4,6,5,**4**,**5**] -> [3,4,6,5,**5**,**6**] -> [3,4,6,5,**6**,**7**] 6th team has maximum points so they be chosen in this case.
Explanation (Test Case 5) Only one team can make it to the playoffs, there is 1 day left of the league stage and 1 point on each day is to be gained. Every team has the probability to qualify for playoffs as their scores are all tied and no team has a score of 1 point. As all 8 teams have the same scores, all of their scenarios will look like this: ALL TEAMS: [**5**, **5**, **5**, **5**, **5**, **5** ,**5** , **5**] -> [**6**, **6**, **6**, **6**, **6**, **6**, **6**, **6**] So, all 8 teams can qualify as their scores are all tied and there's no team with 1 point.

TUTORIAL VIDEO

YouTube Link/s: https://youtu.be/GoZopq-ooKY - Linear Search https://youtu.be/FZgSngp6ZrY - Binary Search

TAKEAWAYS

Name of Member: Venn P. Delos Santos
Contribution to the Group: Coding the program, recording, documentation
Learnings: Learned how to implement Search algorithms such as Linear and Binary.

Name of Member: Marc Christian Tumaneng
Contribution to the Group: Coding the program, recording, documentation
Learnings: I understand the concept and implementations in real world problems of Linear and Binary search algorithms.

Name of Member: John Mark Pajenago
Contribution to the Group: Coding the program, recording, documentation
Learnings:I learned what is the purpose of Linear and Binary search in the real world and how to use it to solve real world problems.