CMPE 300 ANALYSIS OF ALGORITHMS

PROJECT 3

In this project, you will write a program that fulfills the tasks described below, and then fill the necessary parts in the answer sheet provided to you. You will have one file, namely main. Your program will take one argument, that will be either “part1” or “part2”. If the argument is “part1”, the program should output as described under the PART 1, if it is “part2”, then it should create outputs described under the PART 2 of this description.

DESCRIPTION

PART 1

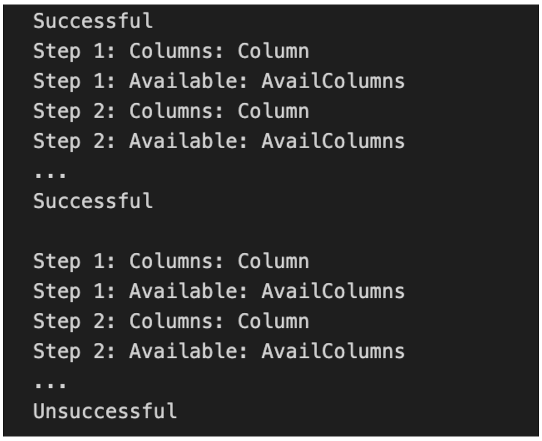
In Part 1, you should output three text files (explained in “a”) and a console output (explained in “b”).

Consider the following Las Vegas algorithm for the *n*-queens problem:

```
function QueensLasVegas ($n$, $Column$ [0:$n$-1])

Input: $n$ (denoting an $n*n$ chess board)
Output: $Column$ [0:$n$-1] (queen $i$ is in row $i$ and column $Column[i]$ such that no
                            two queens attack each other)

$AvailColumns \leftarrow \{0, \ldots, n-1\}$        // all columns are available initially
$R \leftarrow 0$                          // begin with first row
while $AvailColumns \neq \emptyset$ and $R \leq n-1$ do
choose a column $C$ randomly and uniformly from $AvailColumns$
$Column[R] \leftarrow C$              // place queen $R$ in row $R$ and column $C$
$R \leftarrow R+1$
update $AvailColumns$ to be the set of columns available for placing the $(R+1)^{th}$
    queen in row $R+1$ without creating a pair of attacking queens
endwhile

end QueensLasVegas
```

Figure 1

a) Implement the algorithm shown in Figure 1 in any programming language you like. Run your program for *n*=6, n=8, and n=10, and 10000 times for each. For all the runs, the steps of the execution must be shown explicitly in the format:



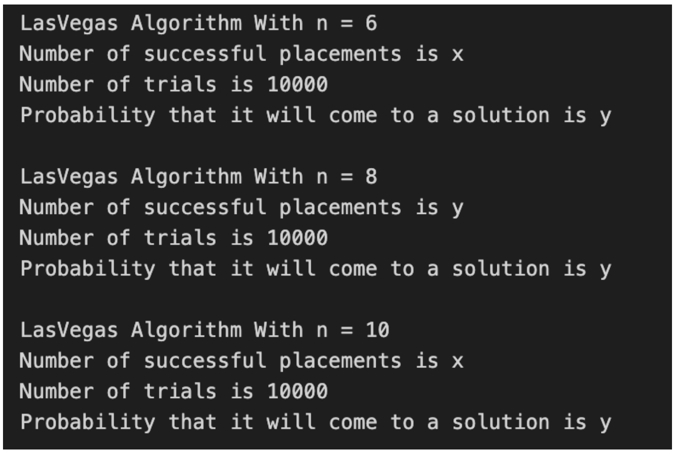
```
Successful
Step 1: Columns: Column
Step 1: Available: AvailColumns
Step 2: Columns: Column
Step 2: Available: AvailColumns
...
Successful

Step 1: Columns: Column
Step 1: Available: AvailColumns
Step 2: Columns: Column
Step 2: Available: AvailColumns
...
Unsuccessful
```

Figure 2

Where Column and AvailColumns are the lists indicated in Figure 1.

For each n value, write the steps into separate files named “results_n.txt” where n is either 6, 8, or 10. *No points will be given if the executions are not shown explicitly in this specified format.*

b) After the execution is finished, your program should write to console the number of successful placements, number of trials, and the probability of successful trials in the format of:

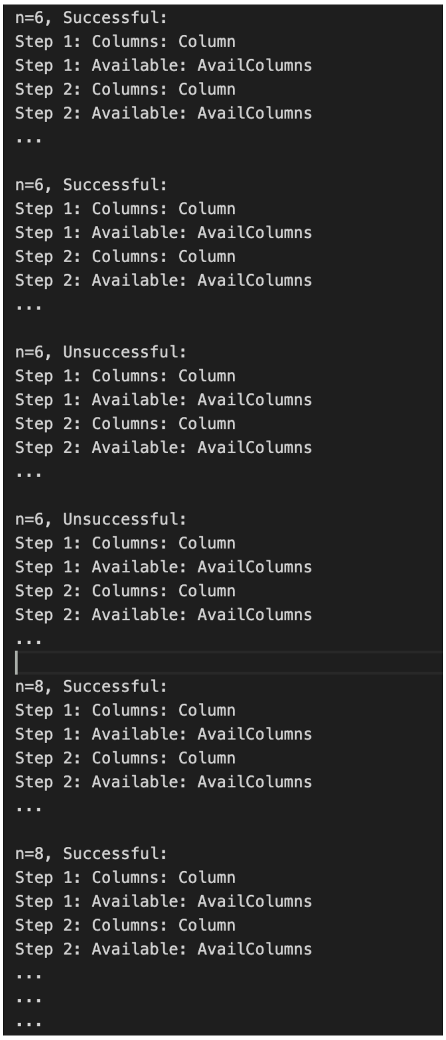


```
LasVegas Algorithm With n = 6
Number of successful placements is x
Number of trials is 10000
Probability that it will come to a solution is y

LasVegas Algorithm With n = 8
Number of successful placements is y
Number of trials is 10000
Probability that it will come to a solution is y

LasVegas Algorithm With n = 10
Number of successful placements is x
Number of trials is 10000
Probability that it will come to a solution is y
```

Figure 3

Where x values are the empirical data you get and y values are (x / 10000)

c) For each n, select two successful and two unsuccessful iterations, copy them into a new file named “results_mini.txt”. Your result_mini.txt file should be in the form of:



```
n=6, Successful:
Step 1: Columns: Column
Step 1: Available: AvailColumns
Step 2: Columns: Column
Step 2: Available: AvailColumns
...

n=6, Successful:
Step 1: Columns: Column
Step 1: Available: AvailColumns
Step 2: Columns: Column
Step 2: Available: AvailColumns
...

n=6, Unsuccessful:
Step 1: Columns: Column
Step 1: Available: AvailColumns
Step 2: Columns: Column
Step 2: Available: AvailColumns
...

n=6, Unsuccessful:
Step 1: Columns: Column
Step 1: Available: AvailColumns
Step 2: Columns: Column
Step 2: Available: AvailColumns
...

n=8, Successful:
Step 1: Columns: Column
Step 1: Available: AvailColumns
Step 2: Columns: Column
Step 2: Available: AvailColumns
...

n=8, Successful:
Step 1: Columns: Column
Step 1: Available: AvailColumns
Step 2: Columns: Column
Step 2: Available: AvailColumns
...
...
...
```

Figure 4

d) Fill out the tables given in the answer sheet under the heading PART 1, “d” with the same data with your “result_mini.txt”

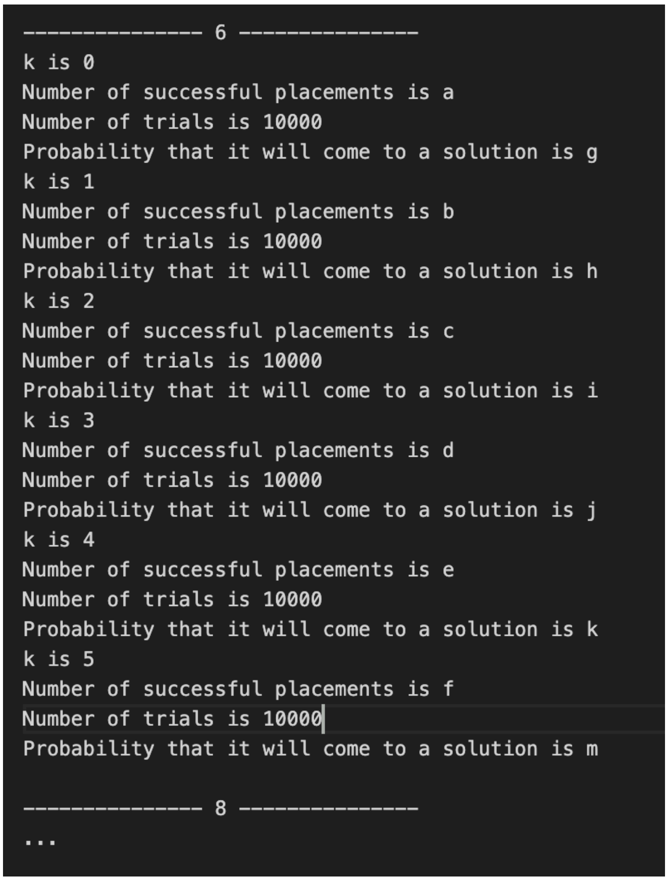
e) Fill out the tables given in the answer sheet under the heading PART 1, “e” with the same data with your console output.

PART 2

In Part 2, you should output a console output (explained in “b”).

a) Write a program similar to QueensLasVegas (Figure 1) that places a number k (k<n) of queens on the board in a random way and then uses backtracking (the deterministic algorithm) to place the remaining queens (without changing the positions of the queens that were placed randomly). That is, after the k queens are fixed, the deterministic algorithm begins from the (k+1)th queen and backtracks when necessary. Test out your program for 10000 times for all values of $k < n$, and for three n values 6, 8, and 10. That means, for n=6, your program should run for 10000 times for k=0, 10000 times for k=1, … 10000 times for k= 5.

b) After the execution is finished, your program should write to console the number of successful placements, number of trials, and the probability of successful trials in the format of:



```
--------------- 6 ---------------
k is 0
Number of successful placements is a
Number of trials is 10000
Probability that it will come to a solution is g
k is 1
Number of successful placements is b
Number of trials is 10000
Probability that it will come to a solution is h
k is 2
Number of successful placements is c
Number of trials is 10000
Probability that it will come to a solution is i
k is 3
Number of successful placements is d
Number of trials is 10000
Probability that it will come to a solution is j
k is 4
Number of successful placements is e
Number of trials is 10000
Probability that it will come to a solution is k
k is 5
Number of successful placements is f
Number of trials is 10000
Probability that it will come to a solution is m

--------------- 8 ---------------
...
```

Figure 5

c) Fill out the tables given in the answer sheet under the heading PART 2, “c” with the same data with your console output.

d) Write comments in the answer sheet under the heading PART 2, “d” about how the success rate changes with k value for each n, and compare the success rates to the original Las Vegas Algorithm.

Notes

Deadline

16.01.2022 Sunday 23:59. Deadline is strict.

No late submission will be accepted.

Submission

You will submit:

- Your main code, which will take one argument, either "part1" or "part2".
- A readme file that describes how to run your code.
- Your 4 result text files for Part 1, namely "results_6.txt", "results_8.txt", "results_10.txt", "results_mini.txt"
- Your answer sheet.

This is a group project. However, each group member must submit the same project with their names. You will name file files as follows (Every member will use only their name and surname):

- Code: main.xxx (xxx: depending on the language)
- Answer Sheet: NameSurname.pdf
- Zip file to upload: NameSurname.zip