

COMS 4996 Parallel Functional Programming
Final Project Proposal – Fall 2021

Deepakraj Dharmapuri Selvakumar
dd3068

Bingo

1 Introduction

Bingo is a game where each player matches numbers called out by the game host in their respective *bingo board*. A Bingo board is a 5 x 5 matrix where each cell has randomly placed unique numbers from 1 to 25. When the game host calls out a number, each player strikes out the number from their board. When the player has a row or a column or any of the diagonals of strikes, then that player wins, if both player gets the strikes at the same time, then it's a draw.

Ex.

Player 1 Board

	B	I	N	G	O
B	18	15	20	10	12
I	22	6	25	3	23
N	7	14	19	2	17
G	8	21	1	9	24
O	16	13	11	5	4

Player 2 Board

	B	I	N	G	O
B	23	13	21	5	14
I	2	15	16	24	20
N	19	18	25	7	4
G	10	9	6	22	11
O	17	3	12	1	8

18	15	20	10	12
22	6	25	3	23
7	14	19	2	17
8	21	1	9	24
16	13	11	5	4

23	13	21	5	14
2	15	16	24	20
19	18	25	7	4
10	9	6	22	11
17	3	12	1	8

Game show calls '6',

Similarly the game show calls a random number for 25 times.

18	15	20	10	12
22	6	25	3	23
7	14	19	2	17
8	21	1	9	24
16	13	11	5	4

23	13	21	5	14
2	15	16	24	20
19	18	25	7	4
10	9	6	22	11
17	3	12	1	8

Player 1 wins since he got a diagonal strikes.

'Bingo!!'

2 Goals

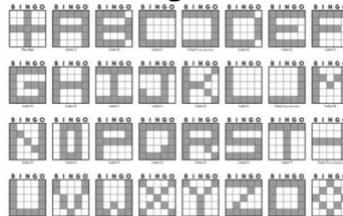
My initial goal is to implement the above board using a 2D array, generate 2 random sequences of 2D array with numbers from 1-25 filled in it. Generate numbers for Game show host and match it for each call the game show makes. We can parallelize a lot of checks, for example, parallelize checks for two boards, individual checks across rows, columns and diagonals. Compare it with serialization algorithm implemented in Haskell along with implementation in Python / C++.

Given enough time, We can introduce different patterns to match like -

- Check from [2][0] to [0][2] and [0][2] to [2][4] diagonally or the mirror of that. Or check for diamond or other shapes.

B	I	N	G	O
		●		
	●		●	
●		Free Space		●
	●		●	
		●		

- Or Introduce Alphabet Pattern Matching etc.,



For checking the above patterns, I am planning to put these patterns in an array data structure or files to introduce more complexity and match the marks on every call among the two players.