Username: kliauw
Name: Karina Fitriani Liauw
Student ID: 1090015

Comp 20003 – Algorithms and Data Structures
Assignment 3 – Experimentation

I. Data from Experiment

Layout 0				
Budget	10.000,00	100.000,00	1.000.000,00	1.500.000,00
Expanded Nodes	2	2	2	2
Generated Nodes	2	2	2	2
Solution Length	2	2	2	2
Number of Pegs Left	1	1	1	1
Expanded/seconds	133333	117647	153846	153846
Time (seconds)	0.000015	0.000017	0.000013	0.000013
Layout 1				
Budget	10.000,00	100.000,00	1.000.000,00	1.500.000,00
Expanded Nodes	3	3	3	3
Generated Nodes	3	3	3	3
Solution Length	3	3	3	3
Number of Pegs Left	1	1	1	1
Expanded/seconds	187500	187500	187500	187500
Time (seconds)	0.000016	0.000016	0.000016	0.000016
Layout 2				
Budget	10.000,00	100.000,00	1.000.000,00	1.500.000,00
Expanded Nodes	7	7	7	7
Generated Nodes	8	8	8	8
Solution Length	6	6	6	6
Number of Pegs Left	1	1	1	1
Expanded/seconds	118644	241379	233333	225806
Time (seconds)	0.000059	0.000029	0.000030	0.000031
Layout 3				
Budget	10.000,00	100.000,00	1.000.000,00	1.500.000,00
Expanded Nodes	3541	3541	3541	3541
Generated Nodes	10282	10282	10282	10282
Solution Length	16	16	16	16

Username: kliauw
Name: Karina Fitriani Liauw
Student ID: 1090015

Number of Pegs Left	1	1	1	1
Expanded/seconds	224156	244004	242584	252441
Time (seconds)	0.015797	0.014512	0.014597	0.014027
Layout 4				
Budget	10.000,00	100.000,00	1.000.000,00	1.500.000,00
Expanded Nodes	1065	1065	1065	1065
Generated Nodes	2418	2418	2418	2418
Solution Length	31	31	31	31
Number of Pegs Left	1	1	1	1
Expanded/seconds	263483	256441	258307	250529
Time (seconds)	0.004042	0.004153	0.004123	0.004251
###				
Layout 5				
Budget	10.000,00	100.000,00	1.000.000,00	1.500.000,00
Expanded Nodes	10000	1000000	1000000	1090275
Generated Nodes	26495	359818	4488464	4898609
Solution Length	32	33	34	35
Number of Pegs Left	**4**	**3**	**2**	**1**
Expanded/seconds	234389	187440	119202	118963
Time (seconds)	0.042664	0.533502	8.389.070	9.164.761
Layout 6				
Budget	10.000,00	100.000,00	1.000.000,00	1.500.000,00
Expanded Nodes	10000	100000	1000000	1500000
Generated Nodes	29368	374378	4481233	7020668
Solution Length	39	40	41	41
Number of Pegs Left	5	4	3	3
Expanded/seconds	241487	195116	121329	115767
Time (seconds)	0.041410	0.512514	8.242.049	12.956.977
Layout 7				
Budget	10.000,00	100.000,00	1.000.000,00	1.500.000,00
Expanded Nodes	10000	100000	1000000	1500000
Generated Nodes	32469	386440	4790308	7173504
Solution Length	34	36	36	36
Number of Pegs Left	4	2	2	2
Expanded/seconds	232612	184588	111344	106286

Username: kliauw
Name: Karina Fitriani Liauw
Student ID: 1090015

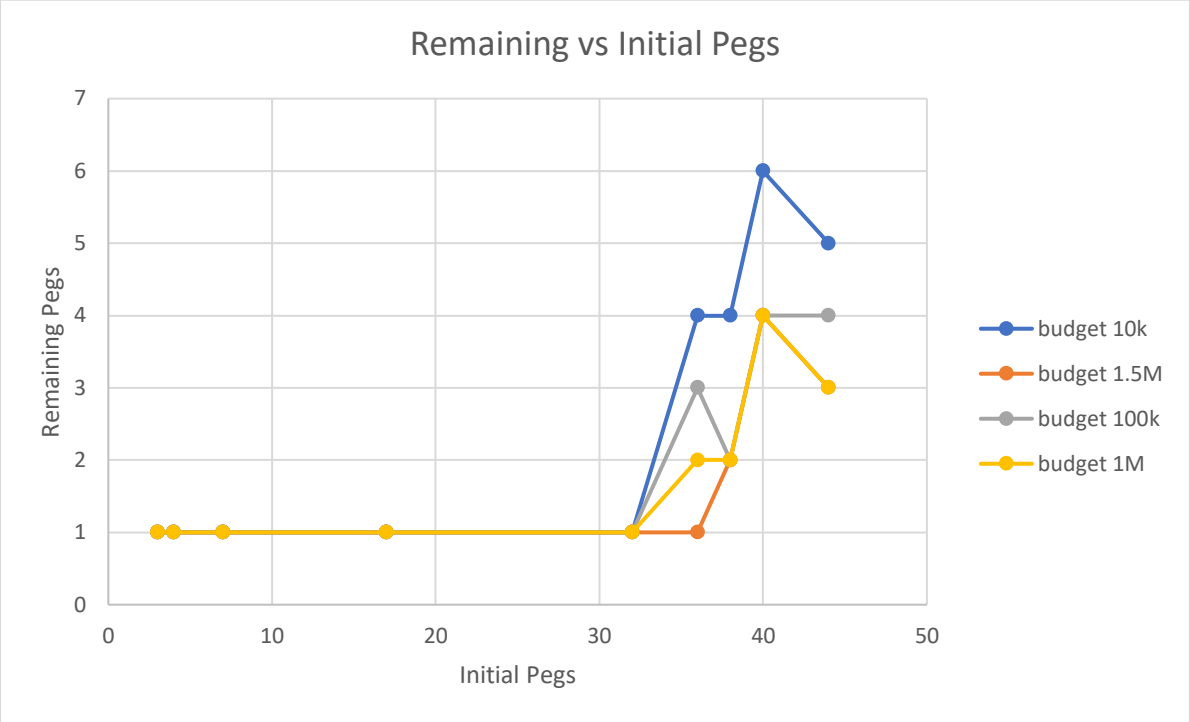
Time (seconds)	0.042990	0.541745	8.981.116	14.112.745
Layout 8				
Budget	10.000,00	100.000,00	1.000.000,00	1.500.000,00
Expanded Nodes	10000	10000	1000000	1500000
Generated Nodes	27562	349921	4073028	6361454
Solution Length	34	36	36	36
Number of Pegs Left	6	4	4	4
Expanded/seconds	241126	191437	112391	106652
Time (seconds)	0.041472	0.522365	8.897.454	14.064.315

II. Explanation of Results

According to the results, the algorithm can solve layout 0 to layout 4 with number of initial pegs ranging from 3-32 considerably well, leaving the number remaining pegs to 1. In terms of time needed to solve the problems, the algorithm also didn't struggle much in solving layout 0 to 4. However, as the number of pegs grow as well as the grid changes, it is noticeable that the algorithm started to struggle both in solving time as well as leaving the remaining pegs to 1 starting from layout 5. In layout 5, there is a noticeable trend that as the budget increases, and when the budget is 1.5000.000 the breadth first search algorithm is able to solve the game. It can be concluded that this algorithm takes more memory as well as time exponentially in order to solve more complex problems (For example, NP Problem).

III. Plot #Pegs Left as a function of #Initial Pegs

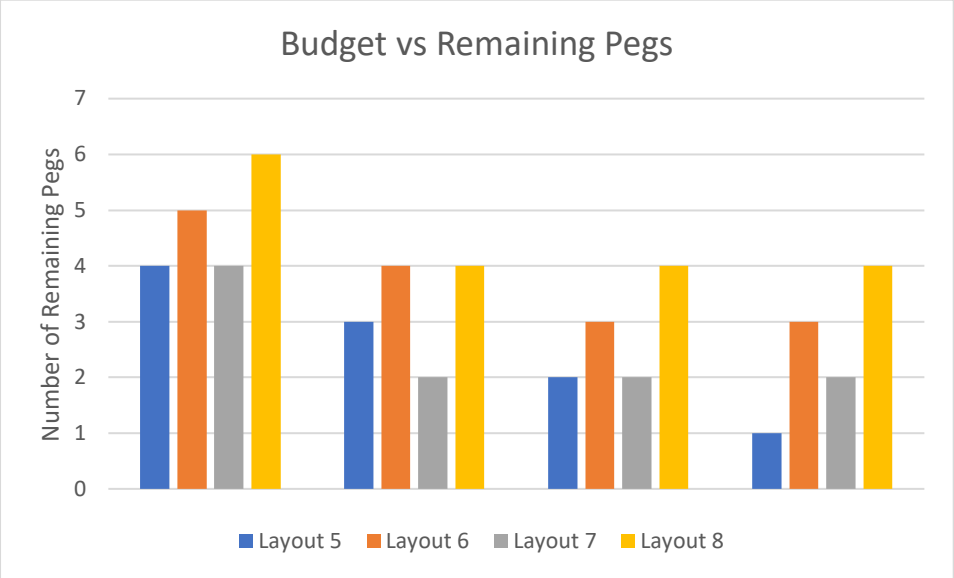
Username: kliauw
Name: Karina Fitriani Liauw
Student ID: 1090015



From the figure above, it can be seen that as the number of initial pegs increase, number of remaining pegs increase because the breadth first search algorithm struggles more in finding the way to solve the problem. Although, some problems may be easier to solve even though the initial pegs are higher, for example layout 8 compared to layout 6, although layout 8 has fewer initial peg, the number of remaining pegs in layout 6 is smaller in the end. According to the plots above, it can be seen that the number of remaining pegs decrease as we increase the budget from 10.000 to maximum of 1.500.000.

IV. Plot Budget Affect Solution Quality (in the last 4 layouts)

Username: kliauw
Name: Karina Fitriani Liauw
Student ID: 1090015



As number of possible solutions in the next state increases, the algorithm started to struggle because it will apply jump to the first state if it is possible. In the animation of play_solution, we can clearly see that if there are not any pegs in its surroundings, the pegs will do multiple moves to jump but not reducing number of remaining pegs. Therefore, it may seem that the algorithm does well in solving the problem, but not accurately optimized to find the best way to solve the problem.

V. Plot of How the Algorithms Works

Below is the pseudocode of the algorithm translated into flowchart.

Username: kliauw
Name: Karina Fitriani Liauw
Student ID: 1090015