Wilfrid Laurier University

Assignment 1

8 Puzzle

October 16, 2022

Nahor Yirgaalem - 190775540 - yirg5540@mylaurier.ca

Shaheer Khan - 190693830 - khan3830@mylaurier.ca

Duong Truong - 190695030 - truo9503@mylaurier.ca

Sameer Mian - 190366140 - mian6140@mylaurier.ca

Memet Rusidovski - 130951550 - rusi1550@mylaurier.ca

CP468: Artificial Intelligence

Dr. Illias kotsireas

Puzzle - H1	Iterations/nodes	# of Moves		H2	Iterations/nodes	# of Moves		H3	Iterations/nodes	# of Moves	
[1, 8, 5, 4, 0, 2, 7, 6, 3]	50	10			15	10			15	10	
[1, 6, 2, 8, 4, 3, 0, 7, 5]	1232	18			386	18			449	18	
[4, 3, 2, 0, 1, 5, 7, 8, 6]	812	17			565	17			415	17	
[1, 3, 0, 7, 2, 4, 5, 8, 6]	39	10			12	10			19	10	
[4, 3, 1, 2, 7, 5, 0, 8, 6]	2878	20			255	20			728	20	
[1, 0, 2, 6, 5, 3, 7, 4, 8]	776	17			196	17			364	17	
[1, 7, 4, 5, 3, 0, 2, 6, 8]	5395	21			295	21			809	21	
[3, 0, 2, 1, 6, 4, 7, 5, 8]	149	13			55	13			52	13	
[4, 2, 7, 5, 1, 6, 8, 0, 3]	11620	23			1691	23			2335	23	
[1, 2, 3, 4, 0, 8, 7, 6, 5]	15	6			8	6			11	6	
[2, 3, 6, 5, 0, 8, 7, 1, 4]	1422	18			283	18			453	18	
[3, 8, 7, 0, 6, 2, 1, 4, 5]	3318	21			158	21			460	21	
[2, 4, 3, 7, 0, 5, 8, 1, 6]	41	10			14	10			16	10	
[5, 6, 1, 4, 2, 0, 7, 8, 3]	692	17			209	17			267	17	
[1, 8, 5, 7, 0, 2, 4, 3, 6]	3084	20			1129	20			973	20	
[1, 2, 5, 7, 4, 6, 0, 8, 3]	541	16			152	16			226	16	
[8, 7, 1, 3, 0, 6, 5, 4, 2]	20545	24			1064	24			2518	24	
[4, 1, 3, 7, 2, 6, 5, 0, 8]	9	7			8	7			8	7	
[2, 5, 0, 7, 3, 8, 1, 4, 6]	385	16			160	16			106	16	
[7, 1, 2, 5, 8, 0, 3, 6, 4]	3362	21			68	21			729	21	
[0, 2, 3, 1, 5, 6, 4, 7, 8]	4	4			4	4			4	4	
[1, 5, 2, 4, 3, 0, 7, 8, 6]	7	5			6	5			6	5	
[7, 1, 5, 3, 6, 2, 4, 8, 0]	2734	20			255	20			566	20	
[2, 1, 8, 4, 6, 3, 0, 7, 5]	1271	18			319	20			347	18	
[5, 1, 3, 0, 2, 7, 4, 8, 6]	66	11			16	11			24	11	
[8, 0, 1, 4, 7, 3, 6, 2, 5]	1432	19			78	19			256	19	
[3, 5, 1, 7, 4, 6, 0, 2, 8]	2735	20			332	20			626	20	
[2, 4, 1, 8, 3, 0, 6, 5, 7]	11309	23			381	23			1972	23	
[1, 2, 3, 7, 4, 6, 5, 0, 8]	7	5			6	5			6	5	
[2, 8, 3, 7, 4, 5, 1, 0, 6]	145	13			63	13			46	13	
[4, 3, 6, 5, 7, 8, 1, 0, 2]	815	17			314	17			202	17	
[6, 4, 1, 7, 5, 3, 2, 8, 0]	5843	22			175	22			1143	22	
[1, 2, 3, 5, 0, 7, 4, 8, 6]	26	8			12	8			15	8	
[1, 3, 0, 4, 2, 5, 7, 8, 6]	4	4			4	4			4	4	

[4, 1, 5, 8, 0, 2, 7, 6, 3]	99	12			14	12			20	12	
[1, 2, 3, 0, 4, 5, 7, 8, 6]	3	3			3	3			3	3	
[6, 1, 3, 2, 5, 8, 7, 4, 0]	1273	18			704	22			343	18	
[2, 6, 0, 1, 7, 3, 8, 4, 5]	1398	18			381	20			417	18	
[5, 2, 7, 1, 6, 3, 0, 8, 4]	5813	22			333	24			1107	22	
[1, 5, 3, 7, 6, 0, 8, 4, 2]	1069	17			210	17			334	17	
[2, 3, 5, 7, 0, 1, 4, 8, 6]	708	16			142	16			209	16	
[5, 2, 0, 3, 7, 1, 4, 8, 6]	2866	20			565	22			697	20	
[3, 6, 4, 0, 5, 7, 2, 8, 1]	10882	23			1276	25			1008	23	
[2, 3, 5, 8, 0, 6, 4, 7, 1]	3129	20			534	20			795	20	
[2, 8, 3, 0, 1, 5, 7, 6, 4]	343	15			120	15			108	15	
[1, 2, 3, 8, 6, 5, 7, 4, 0]	1259	18			227	18			545	18	
[4, 8, 5, 2, 1, 6, 3, 0, 7]	11654	23			523	23			1689	23	
[0, 2, 3, 1, 6, 8, 4, 7, 5]	14	8			8	8			8	8	
[1, 2, 6, 3, 5, 8, 4, 7, 0]	178	14			39	14			88	14	
[0, 3, 4, 1, 6, 2, 7, 8, 5]	392	16			232	18			172	16	
[1, 3, 0, 5, 2, 4, 8, 7, 6]	253	14			51	14			95	14	
[7, 4, 5, 8, 2, 1, 6, 3, 0]	2326	20			89	22			223	20	
[4, 3, 0, 7, 1, 8, 6, 2, 5]	163	14			17	14			30	14	
[4, 2, 3, 5, 0, 6, 7, 1, 8]	139	12			65	12			69	12	
[4, 6, 3, 7, 1, 8, 0, 2, 5]	1322	18			322	18			318	18	
[5, 2, 3, 4, 7, 6, 1, 0, 8]	326	15			97	15			154	15	
[4, 6, 1, 2, 8, 5, 7, 0, 3]	317	15			91	15			89	15	
[1, 3, 5, 2, 4, 6, 8, 0, 7]	334	15			103	15			127	15	
[3, 6, 0, 7, 4, 2, 5, 1, 8]	349	16			57	16			78	16	
[7, 2, 3, 4, 1, 5, 0, 8, 6]	539	16			156	18			209	16	
[2, 6, 8, 1, 3, 0, 4, 5, 7]	134	13			24	13			35	13	
[3, 0, 8, 1, 4, 2, 7, 6, 5]	140	13			19	13			33	13	
[5, 3, 0, 1, 8, 6, 2, 4, 7]	1219	18			381	18			276	18	
[1, 3, 5, 7, 2, 0, 8, 4, 6]	26	9			12	9			12	9	
[1, 5, 2, 7, 0, 6, 8, 3, 4]	119	12			55	12			39	12	
[1, 8, 2, 5, 6, 0, 4, 7, 3]	364	15			102	15			130	15	
[1, 6, 2, 5, 3, 8, 4, 7, 0]	29	10			15	10			14	10	
[7, 1, 2, 5, 3, 0, 8, 4, 6]	138	13			20	13			43	13	
[0, 3, 5, 6, 2, 4, 1, 7, 8]	6177	22			343	22			1459	22	

[8, 1, 2, 5, 3, 0, 4, 7, 6]	147	13			20	13			35	13	
[3, 1, 5, 0, 2, 8, 6, 4, 7]	4874	21			205	21			832	21	
[2, 1, 3, 4, 0, 5, 8, 7, 6]	3435	20			511	20			1446	20	
[0, 1, 8, 5, 3, 6, 4, 7, 2]	1246	18			259	18			277	18	
[1, 3, 5, 0, 7, 2, 8, 4, 6]	41	11			12	11			26	11	
[1, 2, 3, 4, 8, 5, 7, 6, 0]	5	4			4	4			4	4	
[1, 3, 0, 7, 2, 8, 5, 6, 4]	73	12			26	14			26	12	
[1, 2, 6, 5, 3, 7, 0, 4, 8]	388	16			212	16			167	16	
[1, 3, 6, 7, 4, 2, 5, 8, 0]	31	10			17	10			15	10	
[1, 3, 6, 2, 7, 0, 4, 5, 8]	162	13			104	13			76	13	
[4, 5, 1, 7, 6, 2, 0, 8, 3]	203	14			97	14			57	14	
[1, 8, 4, 6, 3, 0, 5, 7, 2]	4911	21			405	21			779	21	
[5, 1, 3, 2, 6, 8, 4, 7, 0]	28	10			13	10			13	10	
[0, 2, 3, 4, 7, 6, 1, 5, 8]	560	16			163	16			220	16	
[1, 5, 2, 4, 6, 8, 0, 7, 3]	40	10			23	10			20	10	
[7, 6, 1, 8, 3, 5, 2, 4, 0]	13326	24			1074	24			2322	24	
[2, 4, 3, 7, 8, 0, 5, 6, 1]	372	15			38	15			73	15	
[4, 0, 2, 7, 1, 3, 8, 5, 6]	22	9			28	9			12	9	
[1, 5, 2, 8, 3, 6, 0, 4, 7]	81	12			39	12			29	12	
[1, 8, 5, 0, 3, 2, 7, 4, 6]	5330	21			996	21			1338	21	
[0, 5, 7, 8, 2, 4, 1, 6, 3]	5624	22			315	22			576	22	
[7, 3, 6, 1, 5, 8, 2, 4, 0]	2627	20			529	22			612	20	
[0, 8, 7, 1, 2, 4, 5, 6, 3]	13028	24			503	24			2193	24	
[2, 7, 3, 1, 8, 5, 6, 0, 4]	682	17			221	21			161	17	
[2, 1, 5, 4, 3, 7, 8, 0, 6]	2421	19			102	19			546	19	
[8, 2, 0, 1, 5, 3, 4, 6, 7]	383	16			65	16			135	16	
[4, 1, 0, 2, 3, 8, 6, 7, 5]	2671	20			288	20			512	20	
[4, 7, 8, 6, 3, 2, 0, 1, 5]	12731	24			845	24			1820	24	
[2, 1, 8, 3, 0, 5, 7, 6, 4]	9433	22			1036	22			1640	22	
[1, 5, 2, 4, 3, 0, 7, 8, 6]	7	5			6	5			6	5	
[5, 7, 2, 1, 3, 0, 8, 4, 6]	134	13			31	13			27	13	
Total runtime of all 100 puzzles was 7.278 seconds					Total runtime of all 100 puzzles was 4.288 s				Total runtime of all 100 puzzles was 6.778 seconds		
Please refer to old tables for average runtimes											

Puzzle - H1	Iterations/nodes	# of Moves		H2	Iterations/nodes	# of Moves		H3	Iterations/nodes	# of Moves	
[1, 6, 2, 4, 10, 3, 11, 7, 13, 5, 9, 8, 14, 15, 0, 12]	216	17			35	17			33	17	
[3, 5, 6, 4, 9, 1, 2, 7, 10, 0, 11, 8, 13, 14, 15, 12]	168	15			28	15			34	15	
[5, 3, 0, 4, 10, 1, 8, 12, 6, 7, 11, 13, 9, 2, 14, 15]	49369	26			638	26			1201	26	
[1, 2, 3, 4, 10, 9, 6, 7, 5, 14, 11, 8, 13, 15, 12, 0]	25	12			18	12			16	12	
[1, 2, 4, 6, 5, 10, 12, 3, 14, 13, 8, 0, 9, 15, 11, 7]	47855	25			654	25			2430	25	
[7, 5, 2, 4, 0, 1, 3, 8, 13, 6, 11, 12, 10, 9, 14, 15]	384	17			69	17			65	17	
[1, 2, 4, 12, 6, 9, 3, 7, 13, 5, 14, 0, 11, 10, 15, 8]	108203	27			396	27			3748	27	
[2, 4, 8, 11, 1, 5, 3, 12, 9, 6, 15, 0, 13, 14, 10, 7]	15185	23			407	23			627	23	
[1, 3, 8, 7, 6, 2, 10, 4, 5, 9, 15, 11, 13, 14, 0, 12]	87	15			57	15			26	15	
[1, 2, 4, 8, 6, 11, 10, 12, 5, 14, 13, 7, 9, 0, 3, 15]	7663	22			1161	24			256	22	
[2, 6, 5, 4, 9, 0, 1, 3, 13, 10, 7, 8, 14, 11, 15, 12]	7621	22			55	24			667	22	
[1, 2, 8, 3, 5, 10, 6, 4, 0, 14, 7, 12, 9, 13, 11, 15]	26	12			15	12			15	12	
[0, 1, 2, 14, 5, 8, 10, 4, 6, 9, 3, 11, 13, 15, 7, 12]	1000001	too long><			5108	34			35967	32	
[1, 2, 8, 3, 5, 10, 4, 7, 15, 13, 14, 6, 0, 9, 12, 11]	25491	25			456	25			1099	25	
[6, 2, 3, 4, 1, 10, 7, 0, 5, 9, 12, 8, 13, 11, 15, 14]	48407	24			3355	28			5519	24	
[1, 2, 8, 3, 9, 0, 15, 6, 14, 5, 4, 11, 13, 10, 7, 12]	1000001	too long><			11739	36			104148	32	
[10, 4, 0, 7, 1, 5, 11, 15, 6, 3, 8, 2, 13, 9, 14, 12]	1000001	too long><			2116	34			42240	34	
[0, 2, 3, 4, 1, 5, 7, 8, 9, 6, 10, 11, 13, 14, 15, 12]	6	6			6	6			6	6	
[2, 10, 3, 4, 0, 5, 7, 8, 1, 9, 13, 12, 14, 6, 11, 15]	14523	23			163	23			1729	23	
[0, 1, 15, 3, 5, 2, 6, 7, 13, 10, 4, 8, 14, 9, 12, 11]	57567	26			286	26			1929	26	
[1, 2, 3, 4, 9, 6, 8, 5, 13, 10, 12, 7, 14, 15, 0, 11]	10289	21			1110	23			1018	21	
[5, 1, 2, 8, 10, 6, 4, 3, 11, 7, 14, 12, 0, 9, 13, 15]	890	21			48	21			60	21	
[2, 5, 3, 4, 13, 0, 6, 8, 9, 1, 14, 11, 10, 15, 7, 12]	26485	24			81	24			1959	24	
[6, 0, 2, 4, 1, 3, 7, 8, 5, 9, 10, 12, 13, 14, 11, 15]	100	13			23	13			43	13	
[2, 6, 3, 4, 1, 13, 11, 7, 5, 0, 10, 8, 14, 9, 15, 12]	104	15			33	15			27	15	
[5, 1, 3, 4, 9, 2, 6, 7, 10, 14, 15, 8, 13, 0, 12, 11]	66	14			14	14			14	14	
[1, 2, 7, 3, 5, 11, 4, 15, 9, 6, 12, 0, 13, 10, 14, 8]	894	17			49	17			93	17	
[5, 1, 7, 6, 3, 2, 0, 4, 9, 14, 11, 12, 13, 15, 8, 10]	126774	27			405	27			5964	27	
[2, 3, 4, 8, 13, 1, 7, 5, 0, 9, 6, 12, 10, 14, 11, 15]	21686	24			374	28			1011	24	
[5, 1, 2, 3, 6, 7, 11, 4, 9, 0, 10, 8, 13, 14, 15, 12]	11	11			11	11			11	11	
[1, 6, 2, 3, 5, 0, 4, 8, 9, 7, 11, 12, 13, 10, 14, 15]	202	14			58	14			96	14	
[14, 6, 0, 4, 2, 3, 13, 11, 5, 8, 15, 7, 9, 1, 10, 12]	1000001	too long><			3820	42			220077	38	
[3, 5, 6, 4, 1, 2, 11, 7, 9, 14, 0, 8, 13, 15, 10, 12]	711	18			70	18			116	18	
[1, 2, 3, 4, 9, 10, 5, 7, 6, 14, 11, 8, 13, 0, 15, 12]	178	14			44	14			86	14	
[3, 2, 4, 8, 1, 13, 7, 12, 5, 0, 10, 15, 11, 9, 6, 14]	112841	27			1961	31			1861	27	
[5, 1, 2, 4, 9, 10, 3, 8, 0, 14, 6, 11, 13, 15, 7, 12]	52	14			16	14			16	14	
[1, 2, 10, 0, 5, 6, 7, 3, 13, 9, 12, 4, 14, 15, 8, 11]	32636	25			218	25			3236	25	
[9, 2, 8, 7, 5, 10, 1, 3, 6, 4, 11, 12, 13, 14, 15, 0]	1000001	too long><			1762	32			75882	32	
[1, 2, 7, 3, 5, 6, 4, 8, 10, 11, 12, 15, 9, 13, 14, 0]	82	14			22	16			35	14	
[1, 3, 7, 0, 5, 2, 8, 4, 9, 6, 11, 12, 13, 10, 14, 15]	9	9			9	9			9	9	
[1, 6, 2, 7, 5, 10, 8, 11, 9, 3, 4, 12, 13, 14, 15, 0]	918	18			24	18			161	18	
[1, 3, 7, 4, 5, 0, 2, 8, 14, 6, 11, 12, 9, 10, 13, 15]	387	16			57	16			145	16	
[0, 1, 7, 4, 2, 3, 12, 8, 6, 9, 14, 15, 5, 13, 10, 11]	3723	22			92	22			191	22	
[1, 7, 2, 4, 10, 3, 8, 12, 6, 9, 13, 11, 5, 0, 14, 15]	6697	22			748	22			377	22	
[1, 2, 3, 4, 9, 5, 7, 8, 0, 6, 10, 12, 13, 14, 11, 15]	6	6			6	6			6	6	
[1, 2, 4, 7, 6, 9, 3, 0, 10, 5, 15, 8, 14, 13, 12, 11]	60034	26			898	26			6214	26	
[3, 2, 4, 8, 0, 1, 6, 12, 5, 9, 11, 7, 13, 10, 14, 15]	928	19			396	19			203	19	
[1, 2, 3, 4, 0, 7, 14, 8, 11, 6, 5, 12, 9, 13, 10, 15]	2550	19			256	19			233	19	
[1, 2, 7, 3, 9, 5, 6, 4, 14, 13, 11, 8, 0, 10, 15, 12]	29	13			44	13			21	13	
[2, 9, 4, 8, 6, 1, 15, 10, 13, 7, 5, 3, 0, 14, 12, 11]	196139	29			118	31			1418	29	
[1, 2, 3, 4, 0, 11, 7, 9, 5, 6, 12, 8, 13, 10, 14, 15]	7397	21			586	21			987	21	
[0, 1, 6, 4, 5, 3, 2, 12, 9, 10, 8, 7, 13, 14, 11, 15]	96	14			21	14			50	14	
[1, 2, 3, 4, 5, 6, 8, 12, 9, 13, 7, 15, 14, 10, 11, 0]	632	16			165	18			208	16	
[5, 1, 2, 3, 9, 10, 6, 4, 13, 14, 12, 8, 11, 7, 0, 15]	466	19			454	19			25	19	
[1, 3, 5, 4, 9, 10, 2, 8, 13, 6, 11, 12, 0, 14, 15, 7]	26340	23			1527	25			2909	23	
[6, 1, 2, 4, 5, 9, 3, 7, 15, 11, 0, 12, 13, 10, 8, 14]	28745	24			199	28			1472	24	
[1, 2, 4, 7, 6, 10, 9, 0, 5, 12, 8, 3, 13, 14, 11, 15]	2956	20			38	20			197	20	
[1, 3, 4, 8, 6, 10, 2, 11, 5, 13, 7, 15, 9, 0, 14, 12]	198	18			193	18			46	18	
[2, 3, 7, 4, 1, 6, 10, 8, 0, 5, 9, 11, 13, 14, 15, 12]	21	12			14	12			14	12	
[1, 2, 3, 4, 6, 8, 11, 7, 9, 13, 14, 12, 0, 10, 5, 15]	57902	25			1337	25			6523	25	
[1, 6, 2, 4, 13, 5, 3, 11, 0, 14, 8, 7, 15, 9, 10, 12]	5085	22			99	22			387	22	
[2, 6, 4, 8, 1, 7, 3, 0, 5, 14, 10, 12, 9, 13, 11, 15]	14	14			14	14			14	14	
[1, 2, 3, 4, 9, 5, 7, 8, 0, 6, 11, 12, 13, 10, 14, 15]	6	6			6	6			6	6	
[6, 1, 8, 7, 3, 0, 2, 11, 5, 9, 12, 4, 13, 10, 14, 15]	3417	22			199	24			172	22	
[1, 2, 7, 3, 10, 0, 6, 4, 5, 9, 14, 8, 13, 15, 11, 12]	76	14			76	14			27	14	
[1, 2, 3, 4, 5, 6, 15, 8, 9, 10, 12, 0, 13, 14, 7, 11]	318	13			83	13			105	13	
[1, 2, 8, 0, 9, 6, 4, 3, 7, 5, 15, 11, 10, 13, 14, 12]	1970	21			59	21			165	21	
[1, 2, 3, 4, 5, 10, 7, 12, 13, 11, 15, 0, 14, 9, 6, 8]	10739	21			609	25			1133	21	
[1, 2, 3, 0, 5, 6, 7, 4, 9, 10, 12, 8, 13, 14, 11, 15]	5	5			5	5			5	5	
[1, 3, 0, 12, 7, 4, 8, 10, 6, 2, 15, 11, 5, 9, 13, 14]	1000001	too long><			27399	40			11460	32	
[1, 0, 3, 4, 5, 2, 6, 8, 9, 11, 7, 12, 13, 15, 10, 14]	587	15			86	15			182	15	
[2, 5, 3, 8, 6, 0, 4, 10, 1, 9, 12, 7, 13, 14, 11, 15]	7106	22			545	24			498	22	
[5, 6, 2, 4, 3, 0, 8, 11, 10, 1, 15, 7, 9, 13, 14, 12]	15959	24			102	24			1415	24	
[9, 5, 3, 4, 13, 1, 11, 7, 14, 2, 10, 8, 6, 0, 15, 12]	1355	20			63	20			43	20	
[5, 4, 3, 8, 9, 2, 0, 11, 13, 6, 15, 7, 10, 1, 14, 12]	112278	27			1004	27			3430	27	
[1, 2, 4, 8, 5, 6, 3, 15, 13, 10, 7, 0, 14, 9, 12, 11]	392	17			32	17			119	17	
[1, 10, 2, 3, 5, 7, 8, 4, 9, 0, 6, 12, 13, 14, 11, 15]	83	13			38	13			30	13	
[2, 6, 3, 4, 1, 7, 11, 9, 5, 0, 13, 8, 14, 15, 10, 12]	4018	21			519	21			227	21	
[1, 2, 4, 8, 5, 3, 12, 0, 10, 6, 7, 15, 9, 13, 14, 11]	186	16			138	16			35	16	

[3, 0, 7, 6, 1, 9, 2, 4, 5, 8, 10, 15, 13, 14, 11, 12]	1000001	too long><			48512	39			533629	35	
[5, 3, 0, 4, 2, 1, 6, 8, 9, 10, 7, 12, 13, 14, 11, 15]	22	10			15	10			14	10	
[1, 2, 8, 3, 5, 6, 11, 7, 13, 14, 0, 12, 10, 4, 9, 15]	129907	26			392	26			6007	26	
[5, 2, 4, 8, 6, 1, 3, 11, 9, 10, 7, 12, 13, 14, 0, 15]	176	15			36	15			46	15	
[5, 1, 6, 4, 0, 3, 11, 7, 9, 2, 8, 12, 13, 14, 15, 10]	9685	21			878	23			707	21	
[0, 5, 1, 4, 9, 6, 11, 7, 2, 10, 3, 15, 13, 14, 12, 8]	201350	28			663	28			7284	28	
[1, 3, 4, 7, 13, 5, 2, 6, 0, 11, 9, 8, 10, 14, 12, 15]	16851	24			309	24			598	24	
[1, 2, 3, 4, 6, 10, 9, 8, 5, 14, 7, 15, 0, 13, 12, 11]	786	17			29	17			130	17	
[1, 2, 3, 4, 10, 6, 8, 12, 5, 14, 0, 11, 13, 7, 15, 9]	24911	22			1226	24			1781	22	
[2, 3, 4, 0, 1, 5, 6, 8, 9, 10, 7, 12, 13, 14, 11, 15]	9	9			9	9			9	9	
[0, 1, 2, 8, 5, 10, 6, 3, 9, 7, 12, 4, 13, 14, 11, 15]	202	16			35	16			45	16	
[1, 0, 6, 4, 9, 5, 2, 3, 13, 11, 15, 7, 14, 10, 12, 8]	427	19			19	19			31	19	
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 15, 11, 13, 0, 14, 12]	4	4			4	4			4	4	
[1, 2, 3, 4, 5, 6, 8, 7, 9, 10, 15, 11, 0, 14, 13, 12]	112553	25			9643	25			19839	25	
[1, 0, 2, 4, 5, 10, 3, 12, 9, 7, 8, 15, 13, 14, 6, 11]	317	15			28	15			32	15	
[1, 2, 3, 4, 0, 5, 6, 8, 14, 10, 13, 12, 9, 11, 7, 15]	332	15			39	15			44	15	
[1, 3, 4, 7, 14, 2, 10, 0, 6, 13, 8, 11, 5, 9, 15, 12]	15192	24			376	28			501	24	
[6, 5, 2, 0, 9, 1, 3, 4, 7, 10, 8, 12, 13, 14, 11, 15]	125096	27			1699	31			11343	27	
[2, 10, 3, 4, 1, 6, 12, 7, 5, 14, 15, 11, 9, 13, 8, 0]	3788	22			104	24			545	22	
[1, 8, 3, 2, 5, 10, 7, 0, 14, 6, 4, 12, 9, 13, 11, 15]	1000001	too long><			9762	34			45666	30	
[1, 10, 2, 4, 9, 8, 0, 12, 5, 3, 6, 15, 13, 7, 11, 14]	1000001	too long><			13714	31			72124	31	
Total runtime of all 100 puzzles was 591.18 seconds					Total runtime of all 100 puzzles was 53.398 seconds				Total runtime of all 100 puzzles was 401.616 seconds		
Please refer to old tables for average runtimes											

Puzzle - H1	Iterations/nodes	# of Moves		H2	Iterations/nodes	# of Moves		H3	Iterations/nodes	# of Moves
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 24, 20, 21, 0, 22, 23]	18502	21			1004	21			1070	21
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 17, 14, 23, 15, 16, 13, 24, 19, 20, 21, 22, 18, 0, 12]	10210	20			2736	20			1938	20
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 22, 17, 20, 0, 24, 21, 18, 23, 19]	205300	28			8099	28			68433	28
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 15, 0, 16, 17, 18, 14, 20, 21, 22, 23, 19, 24]	26402	25			2337	25			8800	25
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 14, 15, 0, 16, 17, 13, 18, 20, 21, 22, 23, 19, 24]	287266	29			11886	29			95755	29
[1, 2, 3, 4, 5, 6, 7, 8, 10, 0, 11, 12, 13, 9, 15, 16, 17, 18, 14, 20, 21, 22, 23, 19, 24]	158225	30			too long	30			52741	30
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 15, 20, 16, 17, 18, 14, 0, 21, 22, 23, 19, 24]	25252	24			1582	24			8417	24
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 15, 0, 16, 17, 18, 14, 20, 21, 22, 23, 19, 24]	26402	25			2405	25			8800	25
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 15, 0, 16, 17, 18, 14, 20, 21, 22, 23, 19, 24]	26402	25			2237	25			8800	25
[1, 2, 3, 4, 5, 6, 7, 8, 9, 0, 11, 12, 13, 14, 10, 16, 17, 19, 24, 15, 20, 21, 18, 22, 23]	319272	30			4351	30			106424	30
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 24, 23, 20, 21, 22, 19, 0]	207433	27			6982	27			69144	27
[1, 2, 3, 4, 0, 6, 7, 8, 9, 5, 11, 12, 13, 15, 10, 16, 17, 18, 14, 20, 21, 22, 23, 19, 24]	26402	27			too long	27			8800	27
[1, 2, 3, 4, 0, 6, 7, 8, 9, 5, 11, 13, 14, 15, 10, 16, 12, 17, 18, 20, 21, 22, 23, 19, 24]	799737	33			48073	33			266579	33
[1, 2, 3, 4, 0, 6, 7, 9, 10, 5, 11, 12, 8, 14, 15, 16, 17, 13, 18, 20, 21, 22, 23, 19, 24]	104361	31			too long	31			34787	31
[1, 2, 3, 4, 5, 6, 7, 8, 9, 0, 11, 12, 13, 15, 10, 16, 17, 18, 14, 20, 21, 22, 23, 19, 24]	26402	26			5686	26			8800	26
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 21, 16, 18, 23, 19, 20, 0, 17, 22, 24]	763594	28			4308	28			254531	28
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 22, 17, 18, 0, 20, 21, 23, 19, 24]	659519	30			3532	30			219839	30
[1, 2, 4, 9, 5, 6, 8, 13, 3, 10, 12, 16, 7, 14, 15, 0, 11, 23, 18, 24, 21, 17, 22, 20, 19]	2253	22			227	22			751	22
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 20, 13, 15, 16, 18, 23, 22, 24, 21, 0, 17, 14, 19]	9215	20			501	20			3071	20
[1, 2, 3, 4, 5, 6, 7, 8, 9, 15, 10, 16, 13, 14, 20, 12, 11, 23, 18, 24, 21, 17, 22, 0, 19]	799456	30			too long	30			266485	30
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 15, 20, 16, 17, 0, 19, 23, 21, 22, 18, 24, 14]	506657	26			10775	26			168885	26
[1, 2, 3, 4, 5, 6, 7, 8, 9, 0, 11, 12, 13, 15, 10, 16, 17, 19, 24, 20, 21, 22, 18, 14, 23]	305957	30			3421	30			101985	30
[1, 2, 3, 4, 5, 6, 7, 8, 9, 0, 11, 12, 14, 22, 10, 16, 19, 18, 17, 15, 21, 23, 13, 24, 20]	24123	26			2444	26			8041	26
[1, 2, 3, 4, 0, 6, 12, 17, 9, 5, 11, 8, 14, 18, 10, 16, 13, 7, 20, 15, 21, 22, 23, 19, 24]	16098	27			788	27			5366	27
[1, 2, 3, 4, 5, 6, 7, 8, 9, 0, 11, 12, 13, 15, 10, 16, 17, 18, 14, 24, 21, 22, 23, 20, 19]	12524	24			1790	24			4174	24
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 17, 12, 14, 15, 21, 16, 13, 18, 0, 20, 22, 23, 19, 24]	196079	28			7593	28			65359	28
[1, 2, 3, 4, 0, 6, 7, 8, 9, 5, 11, 12, 13, 14, 10, 16, 17, 23, 18, 15, 20, 21, 22, 19, 24]	323159	31			9447	31			107719	31
[1, 2, 3, 4, 5, 6, 7, 9, 14, 10, 11, 13, 17, 8, 15, 22, 21, 0, 23, 20, 16, 18, 19, 12, 24]	12170	22			452	22			4056	22
[1, 2, 3, 4, 0, 6, 7, 8, 9, 5, 11, 12, 13, 14, 10, 16, 22, 17, 19, 15, 20, 21, 18, 23, 24]	748621	33			9161	33			249540	33
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 0, 20, 21, 22, 23, 24]	32789	24			1913	24			10929	24
[1, 2, 3, 4, 0, 6, 7, 8, 9, 5, 11, 12, 13, 14, 10, 16, 22, 17, 18, 15, 24, 21, 23, 20, 19]	95738	29			2453	29			31912	29
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 21, 16, 18, 20, 19, 24, 17, 0, 22, 23]	44689	23			19155	23			14896	23
[1, 2, 3, 4, 5, 7, 8, 12, 9, 10, 6, 17, 14, 23, 15, 11, 16, 18, 19, 20, 21, 22, 13, 0, 24]	13321	22			1096	22			4440	22
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 23, 18, 0, 20, 21, 22, 19, 24]	315419	28			9462	28			105139	28
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 0, 21, 16, 23, 18, 15, 20, 17, 22, 19, 24]	24791	25			914	25			8263	25
[1, 2, 3, 4, 5, 6, 7, 8, 9, 0, 11, 12, 13, 14, 10, 21, 16, 18, 19, 15, 20, 17, 22, 23, 24]	691651	32			2081	32			230550	32
[1, 2, 3, 4, 0, 6, 7, 8, 9, 5, 11, 12, 13, 15, 10, 16, 17, 19, 14, 20, 21, 22, 18, 23, 24]	730848	33			3358	33			243616	33
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 20, 23, 19, 24, 21, 22, 18, 0]	752243	29			6128	29			250747	29
[1, 2, 3, 4, 5, 6, 7, 8, 9, 0, 11, 17, 12, 14, 10, 21, 16, 13, 18, 15, 20, 22, 23, 19, 24]	195386	30			7582	30			65128	30
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 17, 21, 19, 20, 0, 16, 22, 18, 23, 24]	2495	20			405	20			831	20
[1, 2, 0, 4, 5, 6, 14, 3, 8, 10, 11, 7, 17, 9, 15, 16, 13, 12, 18, 24, 21, 22, 23, 20, 19]	2025	19			213	19			675	19
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 23, 12, 14, 15, 16, 17, 20, 13, 19, 21, 22, 0, 24, 18]	285616	25			3194	25			95205	25
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 20, 24, 0, 21, 22, 18, 23, 19]	4802	20			1276	20			1600	20

[0, 2, 3, 4, 5, 8, 1, 6, 9, 10, 11, 7, 12, 22, 15, 16, 23, 14, 13, 20, 21, 17, 19, 18, 24]	323902	29			8710	29			107967	29
[1, 2, 3, 4, 5, 6, 7, 9, 14, 0, 11, 13, 8, 15, 10, 16, 12, 17, 19, 23, 21, 22, 18, 24, 20]	too long				46691	41			too long	
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 15, 0, 16, 17, 19, 14, 20, 21, 22, 18, 23, 24]	730848	32			3358	32			243616	32
[1, 2, 3, 4, 5, 6, 7, 8, 9, 0, 11, 12, 13, 14, 10, 16, 21, 17, 24, 15, 20, 19, 18, 22, 23]	762964	30			5993	30			254321	30
[1, 2, 3, 4, 5, 6, 7, 8, 9, 0, 11, 12, 14, 19, 10, 16, 18, 23, 22, 15, 20, 21, 17, 13, 24]	too long				35736	43			too long	
[1, 2, 3, 4, 5, 6, 7, 8, 9, 0, 11, 12, 13, 15, 10, 16, 17, 19, 24, 20, 21, 22, 18, 14, 23]	305957	31			3421	31			101985	31
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 15, 0, 16, 17, 19, 14, 20, 21, 22, 18, 23, 24]	730848	33			3358	33			243616	33
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 24, 15, 22, 17, 20, 0, 16, 21, 18, 23, 19]	too long				72623	43			too long	
[1, 2, 3, 4, 0, 6, 7, 8, 9, 5, 11, 12, 13, 15, 10, 16, 17, 19, 14, 20, 21, 22, 18, 23, 24]	730848	31			3358	31			243616	31
[1, 2, 3, 4, 5, 6, 7, 9, 14, 10, 11, 16, 17, 8, 15, 21, 12, 18, 23, 24, 20, 22, 13, 0, 19]	too long				2656	35			too long	
[1, 2, 3, 4, 0, 6, 7, 8, 9, 5, 11, 12, 13, 15, 10, 16, 17, 18, 20, 19, 21, 22, 14, 24, 23]	274837	28			21037	28			91612	28
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 12, 16, 13, 14, 15, 21, 11, 22, 18, 19, 20, 17, 23, 0, 24]	too long				2400	33			too long	
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 21, 16, 18, 19, 24, 20, 17, 22, 0, 23]	689917	20			2097	20			229972	20
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 19, 18, 15, 16, 17, 14, 24, 23, 21, 22, 20, 0, 13]	11696	28			2506	28			3898	28
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 19, 24, 0, 20, 21, 18, 22, 23]	355651	28			4164	28			118550	28
[1, 8, 4, 9, 5, 7, 3, 13, 2, 10, 6, 11, 14, 0, 15, 17, 21, 18, 12, 24, 16, 22, 23, 20, 19]	too long				2714	35			too long	
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 22, 0, 20, 21, 23, 19, 24]	336126	31			3126	31			112042	31
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 24, 15, 17, 18, 20, 19, 16, 21, 22, 23, 0]	too long				60018	42			too long	
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 0, 21, 16, 18, 19, 15, 20, 17, 22, 23, 24]	773628	28			2050	28			257876	28
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 13, 21, 17, 14, 0, 22, 11, 12, 18, 15, 20, 16, 23, 19, 24]	too long				6543	42			too long	
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 15, 20, 16, 17, 14, 18, 24, 21, 22, 0, 19, 23]	too long				31092	34			too long	
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 17, 21, 18, 19, 0, 20, 16, 22, 23, 24]	too long				5787	37			too long	
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 12, 16, 14, 22, 15, 21, 11, 13, 24, 23, 20, 18, 17, 0, 19]	too long				2120	35			too long	
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 20, 0, 24, 21, 22, 23, 19]	201584	32			4266	32			67194	32
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 16, 11, 12, 17, 0, 13, 23, 18, 20, 15, 21, 22, 19, 14, 24]	too long				4566	36			too long	
[1, 2, 3, 4, 5, 6, 7, 8, 9, 0, 11, 12, 13, 15, 10, 16, 17, 19, 14, 20, 21, 22, 18, 23, 24]	730848	31			3358	31			243616	31
[1, 2, 3, 4, 0, 6, 7, 8, 9, 5, 11, 12, 13, 14, 10, 16, 23, 22, 18, 15, 20, 21, 17, 19, 24]	too long				8473	35			too long	
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 15, 20, 21, 16, 18, 19, 24, 17, 0, 22, 14, 23]	too long				18852	42			too long	
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 15, 0, 16, 17, 19, 14, 20, 21, 22, 18, 23, 24]	730848	33			3358	33			243616	33
[1, 2, 3, 4, 0, 6, 7, 8, 9, 5, 11, 12, 13, 15, 10, 16, 17, 19, 14, 20, 21, 22, 18, 23, 24]	730848	31			3358	31			243616	31
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 19, 21, 17, 13, 15, 22, 11, 14, 20, 24, 12, 16, 0, 18, 23]	too long				3059	42			too long	
[1, 2, 3, 4, 0, 6, 7, 8, 9, 5, 11, 12, 13, 14, 10, 17, 21, 23, 24, 15, 20, 16, 22, 18, 19]	too long				49615	42			too long	
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 15, 0, 16, 23, 18, 14, 20, 21, 17, 22, 19, 24]	665279	32			4115	32			221759	32
[1, 7, 2, 3, 5, 6, 9, 14, 8, 10, 11, 13, 4, 24, 0, 16, 12, 17, 19, 15, 21, 22, 18, 23, 20]	too long				5222	38			too long	
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 0, 16, 22, 17, 19, 15, 20, 21, 18, 23, 24]	too long				8004	34			too long	
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 20, 15, 0, 16, 17, 14, 13, 24, 21, 22, 18, 23, 19]	too long				30181	40			too long	
[1, 2, 3, 4, 5, 6, 7, 8, 9, 0, 11, 13, 17, 14, 10, 21, 16, 12, 20, 15, 24, 22, 18, 23, 19]	666516	32			12645	32			222172	32
[1, 2, 3, 4, 5, 6, 12, 7, 9, 10, 16, 11, 8, 13, 15, 21, 17, 19, 14, 24, 20, 22, 18, 0, 23]	too long				5039	37			too long	
[1, 2, 3, 4, 5, 11, 6, 14, 8, 10, 17, 16, 9, 18, 0, 7, 13, 23, 20, 15, 21, 22, 19, 12, 24]	too long				3979	36			too long	
[1, 2, 3, 4, 5, 6, 7, 8, 9, 0, 11, 12, 13, 15, 10, 16, 17, 19, 14, 20, 21, 22, 18, 23, 24]	730848	32			3358	32			243616	32
[1, 2, 3, 4, 5, 6, 7, 8, 9, 0, 11, 12, 13, 14, 10, 21, 16, 18, 19, 15, 20, 17, 22, 23, 24]	691651	27			2081	27			230550	27
[1, 2, 3, 4, 0, 6, 7, 8, 9, 5, 11, 12, 13, 14, 10, 22, 21, 18, 24, 15, 20, 16, 19, 17, 23]	too long				8894	40			too long	
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 0, 21, 16, 23, 24, 15, 20, 17, 22, 18, 19]	too long				2941	34			too long	
[1, 2, 3, 4, 5, 6, 7, 8, 9, 0, 11, 17, 12, 14, 10, 21, 16, 13, 24, 15, 20, 22, 23, 18, 19]	too long				24356	41			too long	

[1, 2, 3, 4, 5, 11, 6, 8, 9, 0, 7, 12, 13, 14, 10, 22, 21, 17, 18, 15, 20, 16, 23, 19, 24]	too long				6537	39			too long	
[1, 2, 3, 4, 5, 6, 7, 8, 9, 0, 11, 12, 13, 14, 10, 15, 22, 17, 20, 24, 16, 21, 18, 23, 19]	too long				50163	45			too long	
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 19, 13, 15, 16, 22, 17, 23, 0, 20, 21, 18, 24, 14]	too long				55574	41			too long	
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 13, 0, 14, 15, 16, 12, 17, 24, 18, 20, 21, 19, 22, 23]	568418	30			7525	30			189472	30
[1, 2, 3, 4, 5, 7, 11, 8, 9, 10, 6, 16, 22, 13, 15, 20, 21, 12, 18, 0, 24, 23, 17, 14, 19]	too long				79457	49			too long	
[1, 2, 3, 4, 5, 6, 7, 9, 13, 10, 11, 12, 19, 8, 15, 16, 18, 22, 14, 0, 20, 21, 17, 23, 24]	too long				43765	43			too long	
[2, 0, 3, 4, 5, 1, 6, 8, 9, 10, 11, 7, 12, 14, 15, 16, 17, 13, 18, 24, 20, 21, 22, 19, 23]	784650	30			2289	30			261550	30
[1, 2, 3, 4, 5, 6, 7, 8, 9, 0, 11, 12, 14, 18, 10, 16, 22, 17, 13, 15, 20, 21, 23, 19, 24]	too long				20705	39			too long	
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 21, 18, 17, 24, 0, 20, 16, 23, 22, 19]	too long				28893	39			too long	
[1, 2, 3, 4, 5, 6, 7, 9, 13, 10, 11, 19, 8, 15, 20, 22, 12, 0, 17, 24, 16, 21, 18, 14, 23]	too long				13424	41			too long	
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 17, 21, 0, 19, 24, 20, 16, 22, 23, 18]	too long				8247	35			too long	
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 15, 20, 16, 17, 19, 14, 0, 21, 22, 18, 23, 24]	762959	32			3358	32			254319	32
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 13, 17, 15, 0, 16, 12, 19, 14, 20, 21, 22, 18, 23, 24]	too long				21204	38			too long	
Total runtime of all 100 puzzles was ~1 hour					Total runtime of all 100 puzzles was 681.6 s				Total runtime of all 100 puzzles wa	
Please refer to old tables for average runtimes										

CP468 8 Puzzle Assignment

Iteration and expanded node are equivalent in this report

8 Puzzle

Averages of H1: Misplaced Tiles

Iterations	Moves	Time	Number of Puzzles
318	15	500 ms	15
2500	20	3 sec	30
6500	22	13 sec	40
11,000	24	33 sec	13
200,000	31	45 sec	2

Averages of H2: Manhattan Distance

Iterations	Moves	Time	Number of Puzzles
141	18	300 ms	15
375	20	400 ms	30
450	22	500 ms	40
1000	24	2.2 sec	13
9000	31	17 sec	2

Averages of H3: Euclidean Distance

Iterations	Moves	Time	Number of Puzzles
160	16	250 ms	15
310	18	270 ms	30
1300	22	600 ms	40
2000	24	1 sec	13
37000	31	100 sec	2

15 Puzzle

Averages of H1: Misplaced Tiles

Iterations	Moves	Time	Number of Puzzles
80	15	150 ms	3
1,700	21	250 ms	20
7,000	22	450 ms	45
14,000	24	840 ms	30
420,000	28	23 sec	2

Averages of H2: Manhattan Distance

Iterations	Moves	Time	Number of Puzzles
182	21	200 ms	20
11,800	28	4 sec	75
196,000	48	66 sec	2
490,000	56	250 sec	2
560,000	66	265 sec	1

Averages of H3: Euclidean Distance

Iterations	Moves	Time	Number of Puzzles
180	19	325 ms	30
600	22	700 ms	40
1800	24	7 sec	26
12,000	28	27 sec	3
43,000	32	44 sec	1

24 Puzzle

Averages of H1: Misplaced Tiles

Iterations	Moves	Time	Number of Puzzles
38	11	200 ms	17
8400	19	500 ms	40
11,300	20	1.2 sec	40
156,000	25	9.6	2
180,000	26	10.9 sec	1

Averages of H2: Manhattan Distance

Iterations	Moves	Time	Number of Puzzles
70	22	150 ms	17
233	25	350 ms	40
850	27	500 ms	40
60,000	32	31 sec	2
196000	48	66 sec	1

Averages of H3: Euclidean Distance

Iterations	Moves	Time	Number of Puzzles
100	20	300	17
600	22	500 ms	40
3,300	27	1.9 sec	40
184,000	30	100 sec	2
195,000	36	108 sec	1

Comparing H1, H2 & H3

Out of all heuristics the one that requires the least number of iterations is H2. The first heuristic is better than a blind search such as Breadth First Search (BFS), but requires many more iterations to get an answer. The run time of H1 is much faster since time is saved on calculating the cost. These savings in computing time are negated by the increased memory and iterations required. H3 performs similarly to H2, but H2 requires less iterations on average. The adding cost of squaring and taking the root for H3 does not increase the runtime. Most runtime differences came from optimizing the Python code, Iterations were still higher in H3.

Our Third Heuristic

Our third heuristics is the Euclidean Distance. This is calculated by taking the horizontal and vertical number of moves for a tile to be put in the right place. Each number is squared and added together. Then the square root is taken which gives us our H3 value.

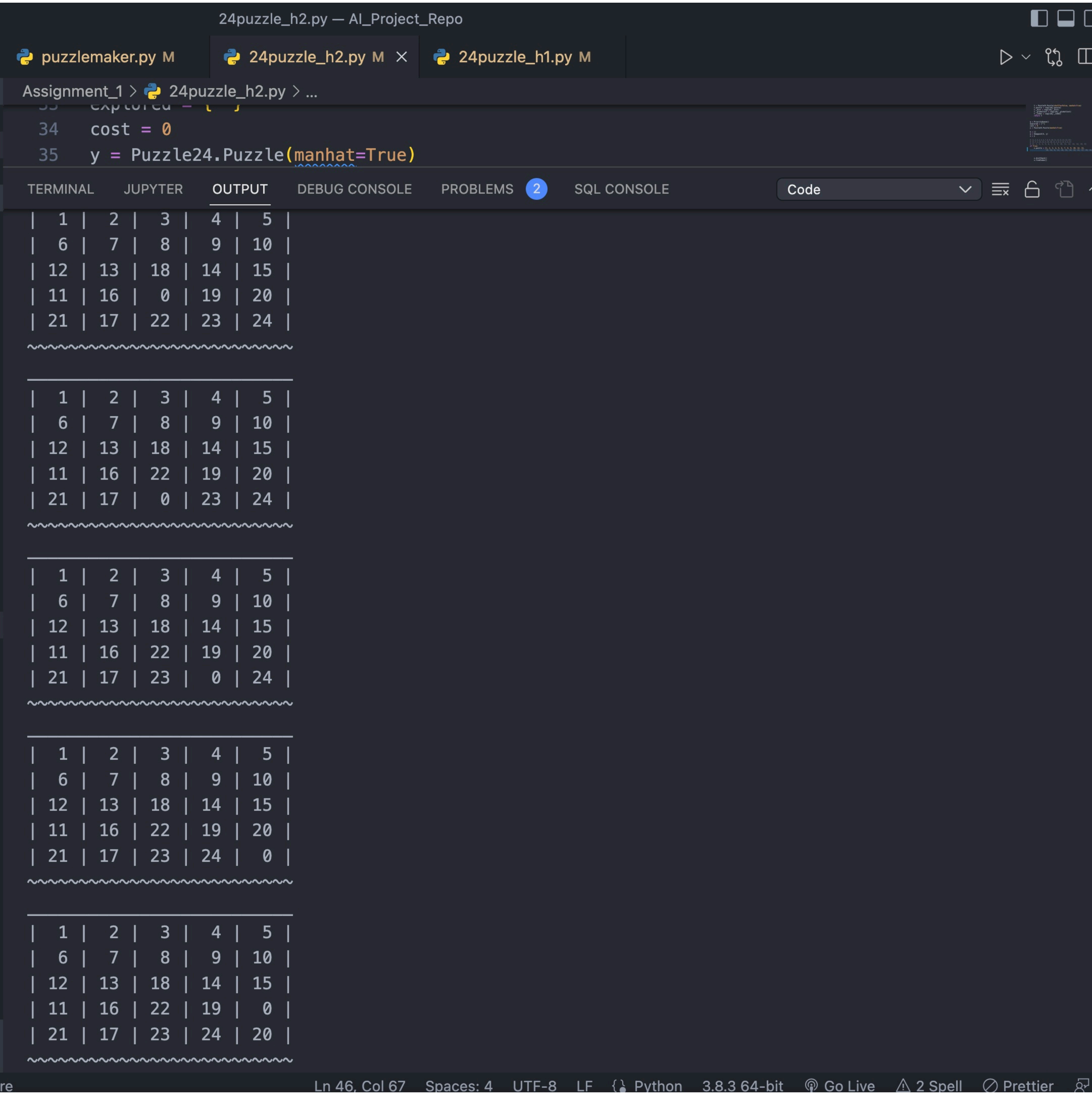
Why is h3 an admissible heuristic?

The euclidean distance never over-estimates the cost of moving a tile to its correct spot because by Triangle Inequality the distance/cost given by H3 will always be less than the number of moves it would take to get to its spot. For example, if a tile needs to move one to the left and one up the euclidean distance will be sqrt(2) which is always less than the total moves needed of 2. At worst the tile needs to move in a straight line and in that case euclidean distance and moves are equal, but still making the heuristic admissible.

```
from puzzles import Puzzle8
from copy import deepcopy
from queue import PriorityQueue
'''
h1 = the number of misplaced tiles. For Figure 3.28,
all of the eight tiles are out of position, so the
start state would have h1 = 8. h1 is an admissible
heuristic because it is clear that any tile that is
out of place must be moved at least once.
'''
"""
puzzles =[]
for i in range(100):
    puzzles.append(Puzzle8.Puzzle())
"""


q = PriorityQueue()
explored = {""}
cost = 0
y = Puzzle8.Puzzle()
z = Puzzle8.Puzzle(shuffle=True)



x = y

if False:
    x.puzzle = [8, 6, 7, 2, 5, 4, 3, 0, 1]#[5, 1, 4, 6, 3, 8, 0, 7, 2]  # [3, 6, 2, 5, 0, 7,
4, 1, 8]
    x.distCheck()
    x.findIndex()

explored.add(str(x.puzzle))

while x._dist != 0 and cost < 2000000:
    up = deepcopy(x)
    down = deepcopy(x)
    left = deepcopy(x)
    right = deepcopy(x)

    x1 = up.up()
    x2 = down.down()
    x3 = left.left()
    x4 = right.right()

    if x1 and str(up.puzzle) not in explored:
        q.put(up)
        explored.add(str(up.puzzle))
        up.parent_node = x
        #print(up, up.puzzle, "up", up._index, "index", "\n\n")
    if x2 and str(down.puzzle) not in explored:
        q.put(down)
        explored.add(str(down.puzzle))
        down.parent_node = x
        #print(down, down.puzzle, "down", down._index, "index", "\n\n\n")
    if x3 and str(left.puzzle) not in explored:
        q.put(left)
        explored.add(str(left.puzzle))
        left.parent_node = x
        #print(left, left.puzzle, "left", left._index, "index", "\n\n\n")
    if x4 and str(right.puzzle) not in explored:
        q.put(right)
        explored.add(str(right.puzzle))
        right.parent_node = x
        #print(right, right.puzzle, "right", right._index, "index", "\n\n")
    x = q.get()
    x._globalCost += 1

    if cost % 100 == 0:
        print(cost)
    #print(x._dist, " -------", x._globalCost)
    cost += 1


temp = x
lst = []
while temp.parent_node != None:
    lst.append(temp)
    temp = temp.parent_node

for i in lst:
    print(i)


print(x._globalCost)
print(cost)
```

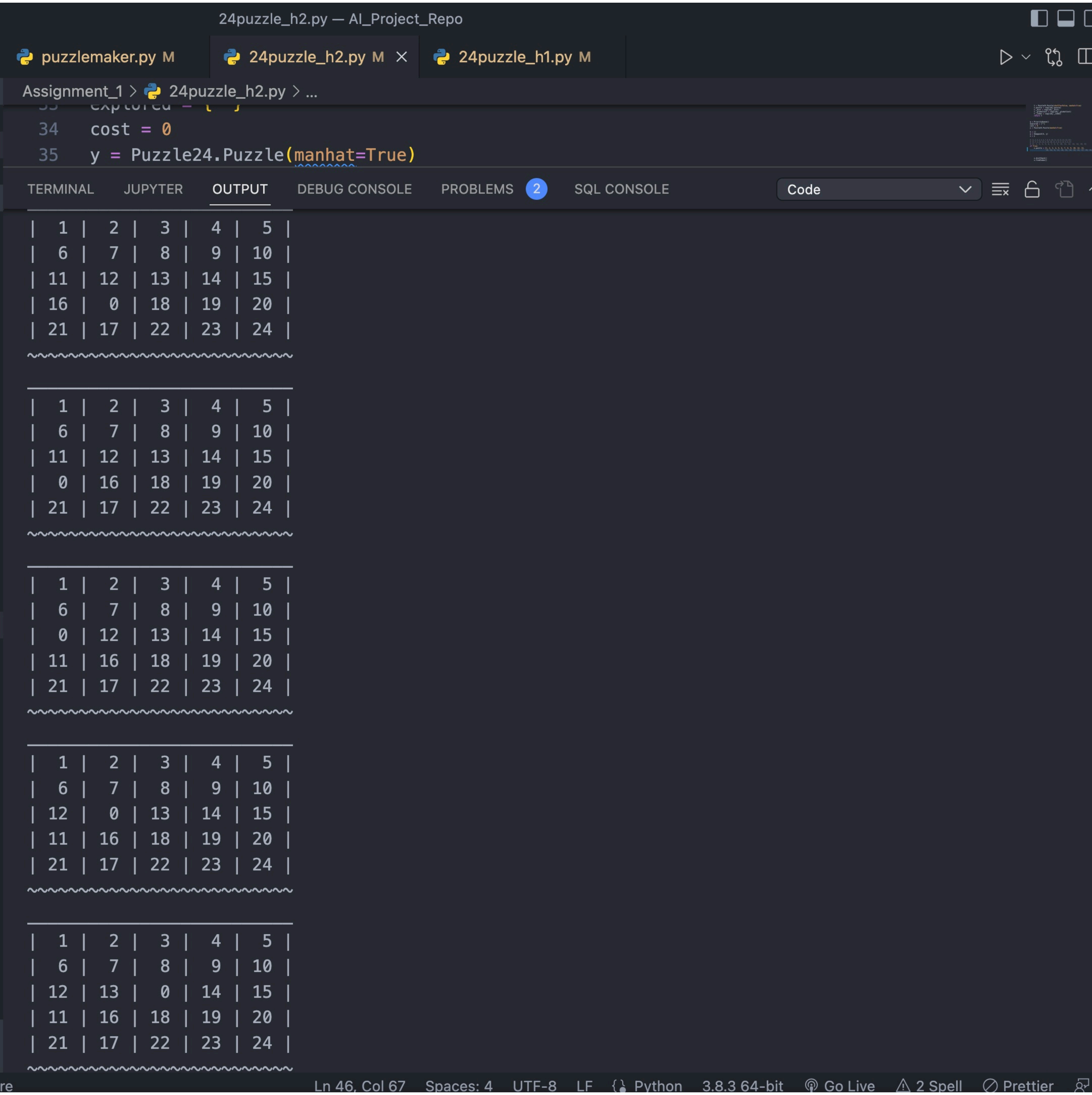
```python
import math
import random
import numpy as np

'''
Zero is the place holder for the empty square.
the matrix is divided equally so,
[1,2,3,4,5,6,7,8, 0] =>

    __________
    |1 | 2| 3|
    |4 | 5| 6|
    |7 | 8| 0|
    ~~~~~~~~~~
'''


class Puzzle:

    def __init__(self, size=3, shuffle=True, manhat=False, ecd=False):
        self.size = size
        self.puzzle = []  # [1, 2, 3, 4, 5, 6, 7, 8, 0]
        self.createPuz(size)
        self._index = 8
        self._dist = 0
        self._solved = False
        self._globalCost = 0
        self.parent_node = None
        self._manhat=manhat
        self._ecd = ecd

        if(shuffle):
            self.scramble()
            self.distCheck()

    def createPuz(self, size):
        for x in range(1, size*size):
            self.puzzle.append(x)
        self.puzzle.append(0)

    def __str__(self):
        return "_____________\n| {0} | {1} | {2} |\n" \
            "| {3} | {4} | {5} |\n| {6} | {7} | {8} |\n~~~~~~~~~~~~~".format(
                *self.puzzle)

    def findIndex(self):
        i = 0
        for x in range(9):
            if self.puzzle[x] == 0:
                i = x
                self._index = i
            #print(self.puzzle[x], "{\}", end="")

        return i

    def scramble(self):
        random.shuffle(self.puzzle)
        self.findIndex()

    def distCheck(self):
        dist = 0
        if self._manhat:
            g1 = np.asarray(self.puzzle).reshape(3, 3)
            g2 = np.asarray([1, 2, 3, 4, 5, 6, 7, 8, 0]).reshape(3, 3)

            for i in range(8):
                a, b = np.where(g1 == i+1)
                x, y = np.where(g2 == i+1)
```

```
                dist += abs((a-x)[0])+abs((b-y)[0])

        if self._ecd:
            g1 = np.asarray(self.puzzle).reshape(3, 3)
            g2 = np.asarray([1, 2, 3, 4, 5, 6, 7, 8, 0]).reshape(3, 3)

            for i in range(8):
                a, b = np.where(g1 == i+1)
                x, y = np.where(g2 == i+1)
                dist +=  math.sqrt((abs((a-x)[0]) ** 2) + (abs((b-y)[0]) ** 2))
        else:
            for i, j in zip(self.puzzle, range(9)):
                if i != (j + 1) and (i != 0 ):
                    dist += 1

        self._dist = dist
        return dist

    def up(self):
        if(0 in self.puzzle[((self.size ** 2)-self.size):]):
            #print("in bottom: invalid")
            return False
        else:
            self.puzzle[self._index], self.puzzle[self._index +
                                                  3] = self.puzzle[self._index + 3],
self.puzzle[self._index]
            self.distCheck()
            self.findIndex()
            #print(self._index,"......")
            return True

    def down(self):
        if(0 in self.puzzle[0:self.size]):
            #print("in top: invalid")
            return False
        else:
            self.puzzle[self._index], self.puzzle[self._index -
                                                  3] = self.puzzle[self._index - 3],
self.puzzle[self._index]
            self.distCheck()
            self.findIndex()
            return True

    def right(self):
        if (self._index != 0 and self._index != 3 and self._index != 6):
            #swap the index to the left
            self.puzzle[self._index], self.puzzle[self._index -
                                                  1] = self.puzzle[self._index - 1],
self.puzzle[self._index]
            self.distCheck()
            self.findIndex()
            return True
        else:
            #print("Invalid Move")
            return False

    def left(self):
        if (self._index != 2 and self._index != 5 and self._index != 8):
            #swap the index to the right
            self.puzzle[self._index], self.puzzle[self._index +
                                                  1] = self.puzzle[self._index + 1],
self.puzzle[self._index]
            self.distCheck()
            self.findIndex()
            return True
        else:
            #print("Invalid Move")
            return False

    def __iter__(self):
        for v in self.puzzle:
            yield v

    def __lt__(self, obj):
        return (self._dist + self._globalCost) < (obj._dist + obj._globalCost)


'''
Testing
x = Puzzle()


print(x.findIndex())

print(x)
x.down()
print(x)

'''
```



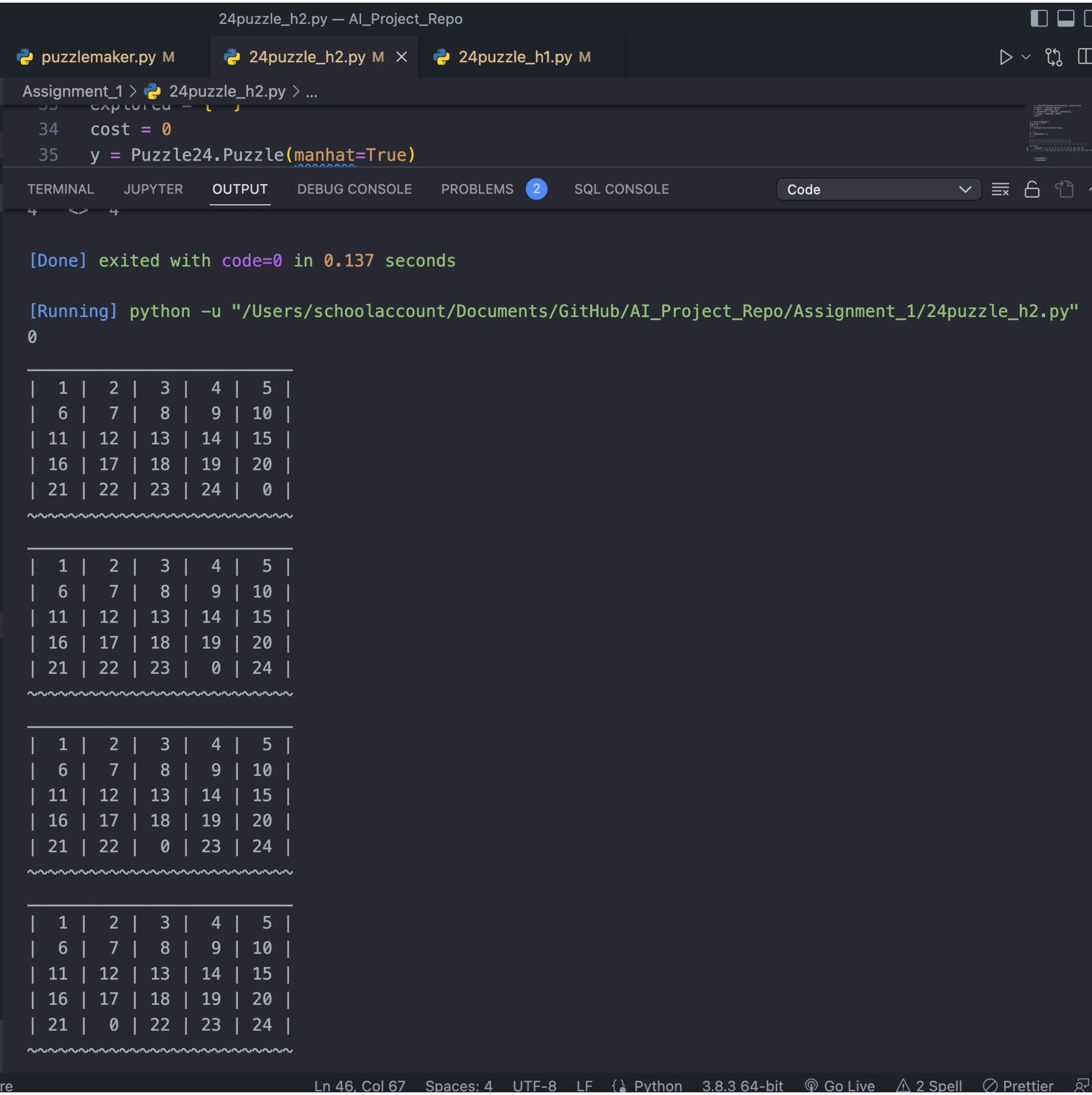
```
explored = [ ]
cost = 0
y = Puzzle24.Puzzle(manhat=True)
```

```
|  1 |  2 |  3 |  4 |  5 |
|  6 |  7 |  8 |  9 | 10 |
| 12 | 13 | 18 | 14 | 15 |
| 11 | 16 |  0 | 19 | 20 |
| 21 | 17 | 22 | 23 | 24 |
~~~~~~~~~~~~~~~~~~~~~~~~~~

__________________________
|  1 |  2 |  3 |  4 |  5 |
|  6 |  7 |  8 |  9 | 10 |
| 12 | 13 | 18 | 14 | 15 |
| 11 | 16 | 22 | 19 | 20 |
| 21 | 17 |  0 | 23 | 24 |
~~~~~~~~~~~~~~~~~~~~~~~~~~

__________________________
|  1 |  2 |  3 |  4 |  5 |
|  6 |  7 |  8 |  9 | 10 |
| 12 | 13 | 18 | 14 | 15 |
| 11 | 16 | 22 | 19 | 20 |
| 21 | 17 | 23 |  0 | 24 |
~~~~~~~~~~~~~~~~~~~~~~~~~~

__________________________
|  1 |  2 |  3 |  4 |  5 |
|  6 |  7 |  8 |  9 | 10 |
| 12 | 13 | 18 | 14 | 15 |
| 11 | 16 | 22 | 19 | 20 |
| 21 | 17 | 23 | 24 |  0 |
~~~~~~~~~~~~~~~~~~~~~~~~~~

__________________________
|  1 |  2 |  3 |  4 |  5 |
|  6 |  7 |  8 |  9 | 10 |
| 12 | 13 | 18 | 14 | 15 |
| 11 | 16 | 22 | 19 |  0 |
| 21 | 17 | 23 | 24 | 20 |
~~~~~~~~~~~~~~~~~~~~~~~~~~
```



```
    explored = [ ]
    cost = 0
    y = Puzzle24.Puzzle(manhat=True)
```

```
|  1 |  2 |  3 |  4 |  5 |
|  6 |  7 |  8 |  9 | 10 |
| 11 | 12 | 13 | 14 | 15 |
| 16 |  0 | 18 | 19 | 20 |
| 21 | 17 | 22 | 23 | 24 |
~~~~~~~~~~~~~~~~~~~~~~~~~~

__________________________
|  1 |  2 |  3 |  4 |  5 |
|  6 |  7 |  8 |  9 | 10 |
| 11 | 12 | 13 | 14 | 15 |
|  0 | 16 | 18 | 19 | 20 |
| 21 | 17 | 22 | 23 | 24 |
~~~~~~~~~~~~~~~~~~~~~~~~~~

__________________________
|  1 |  2 |  3 |  4 |  5 |
|  6 |  7 |  8 |  9 | 10 |
|  0 | 12 | 13 | 14 | 15 |
| 11 | 16 | 18 | 19 | 20 |
| 21 | 17 | 22 | 23 | 24 |
~~~~~~~~~~~~~~~~~~~~~~~~~~

__________________________
|  1 |  2 |  3 |  4 |  5 |
|  6 |  7 |  8 |  9 | 10 |
| 12 |  0 | 13 | 14 | 15 |
| 11 | 16 | 18 | 19 | 20 |
| 21 | 17 | 22 | 23 | 24 |
~~~~~~~~~~~~~~~~~~~~~~~~~~

__________________________
|  1 |  2 |  3 |  4 |  5 |
|  6 |  7 |  8 |  9 | 10 |
| 12 | 13 |  0 | 14 | 15 |
| 11 | 16 | 18 | 19 | 20 |
| 21 | 17 | 22 | 23 | 24 |
~~~~~~~~~~~~~~~~~~~~~~~~~~
```



```
explored = [ ]
cost = 0
y = Puzzle24.Puzzle(manhat=True)
```

```
4  <>  4

[Done] exited with code=0 in 0.137 seconds

[Running] python -u "/Users/schoolaccount/Documents/GitHub/AI_Project_Repo/Assignment_1/24puzzle_h2.py"
0
 ____________________________
|  1 |  2 |  3 |  4 |  5 |
|  6 |  7 |  8 |  9 | 10 |
| 11 | 12 | 13 | 14 | 15 |
| 16 | 17 | 18 | 19 | 20 |
| 21 | 22 | 23 | 24 |  0 |
~~~~~~~~~~~~~~~~~~~~~~~~~~~~
 ____________________________
|  1 |  2 |  3 |  4 |  5 |
|  6 |  7 |  8 |  9 | 10 |
| 11 | 12 | 13 | 14 | 15 |
| 16 | 17 | 18 | 19 | 20 |
| 21 | 22 | 23 |  0 | 24 |
~~~~~~~~~~~~~~~~~~~~~~~~~~~~
 ____________________________
|  1 |  2 |  3 |  4 |  5 |
|  6 |  7 |  8 |  9 | 10 |
| 11 | 12 | 13 | 14 | 15 |
| 16 | 17 | 18 | 19 | 20 |
| 21 | 22 |  0 | 23 | 24 |
~~~~~~~~~~~~~~~~~~~~~~~~~~~~
 ____________________________
|  1 |  2 |  3 |  4 |  5 |
|  6 |  7 |  8 |  9 | 10 |
| 11 | 12 | 13 | 14 | 15 |
| 16 | 17 | 18 | 19 | 20 |
| 21 |  0 | 22 | 23 | 24 |
~~~~~~~~~~~~~~~~~~~~~~~~~~~~
```