



MARMARA UNIVERSITY
FACULTY OF ENGINEERING
COMPUTER ENGINEERING DEPARTMENT

CSE 4082 - Assignment 1

Group Members

Ezgi Eren – 150119515

Emre Cihan Varlı – 150119711

İrem Kubalas – 150119721

1. Introduction

The Knight's Tour Problem (KTP) involves finding a sequence of moves for a knight on a chessboard such that the knight visits every square exactly once. This project implements multiple search strategies to solve KTP efficiently, including uninformed and informed methods. Each method is tested on varying board sizes to analyze its performance and effectiveness.

2. Problem Definition and Search Space

The problem is formally defined as:

- **State Space:** A sequence of visited squares on an $n \times n$ chessboard.
- **Operators:** Legal knight moves.
- **Goal Test:** All squares on the board are visited exactly once.
- **Path Cost:** Zero (goal state depends only on the final state).

3. Search Methods Implemented

The implemented algorithms use the **Tree Search** framework, a foundational approach in artificial intelligence, to explore the solution space systematically. Each method constructs a search tree where nodes represent board states and edges represent knight moves.

In our code, the Tree Search algorithm is utilized, in the TreeSearch.java file lines 9 to 75, as the backbone for systematically exploring all possible sequences of knight moves. The framework involves the following key elements:

- **Node Representation:** Each node represents a unique state of the chessboard, including the knight's position and the set of visited squares.
- **Edge Representation:** Each edge represents a valid knight move from one square to another.
- **Search Strategies:** BFS, DFS, and heuristic-based searches dictate the order in which nodes are expanded.

The Tree Search algorithm enabled us to systematically explore the vast search space of the Knight's Tour Problem while leveraging heuristics to optimize performance. Specifically:

- **BFS and DFS** utilized Tree Search to exhaustively explore all possible paths, ensuring completeness.
- **Warnsdorff's heuristic (h1b)** optimized the node expansion process by prioritizing moves with the fewest available options, indirectly preventing dead-ends.
- The improved heuristic (h2) further enhanced the Tree Search algorithm by introducing a tie-breaking rule, enabling it to handle larger board sizes effectively.

4. Result and Analysis

4.1 Test results for N=8

4.1.1 BFS (Breadth-First Search) Result

```
erene@DESKTOP-L4U7C0A MINGW64 ~/OneDrive/Masaüstü/ai_project1/ai_pro1/src
$ java -Xmx4g -Xss1g Main 8 BFS 15
Exception in thread "main" java.lang.OutOfMemoryError: Java heap space
    at java.base/java.util.Arrays.copyOfRange((Arrays.java:3852)
    at java.base/java.lang.String.<init>(String.java:4889)
    at java.base/java.lang.StringBuilder.toString(StringBuilder.java:478)
    at TreeSearch.boardToString(TreeSearch.java:259)
    at TreeSearch.search(TreeSearch.java:60)
    at Main.main(Main.java:16)
```

- Received a memory failure error (OutOfMemoryError: Java heap space).
- The reason for the error: the BFS needs to save all Decrement states to memory. even for an 8x8 board, memory becomes insufficient because the possible situations are too many.
- This shows that BFS is Decently disadvantaged in terms of memory in large search spaces.

4.1.2 DFS (Depth-First Search) Result

```
erene@DESKTOP-L4U7C0A MINGW64 ~/OneDrive/Masaüstü/ai_project1/ai_pro1/src
$ java -Xmx4g -Xss1g Main 8 DFS 15
Exception in thread "main" java.lang.OutOfMemoryError: Java heap space
    at TreeSearch.generateSuccessors(TreeSearch.java:272)
    at TreeSearch.search(TreeSearch.java:58)
    at Main.main(Main.java:16)
```

- DFS also gave a memory error in a similar way .
- However, the error occurred at a different point (in the generateSuccessors method) than in BFS.
- This indicates that the DFS has reached its memory limit while trying to dig deeper.
- While it is expected to use less memory than BFS, it is still insufficient for 8x8.

4.1.3 DFS-H1B (Warnsdorff's Rule) Result

Solution								-	□	×	
a	b	c	d	e	f	g	h				
15	30	11	50	25	28	9	62				
12	49	14	29	10	63	24	27				
31	16	51	46	53	26	61	8				
48	13	54	35	60	37	44	23				
17	32	47	52	45	58	7	38				
2	55	34	59	36	41	22	43				
33	18	1	4	57	20	39	6				
0	3	56	19	40	5	42	21				

```
erene@DESKTOP-L4U7C0A MINGW64 ~/OneDrive/Masaüstü/ai_project1/ai_pro1/src
$ java -Xmx4g -Xss1g Main 8 DFS-H1B 15
Search Method: DFS-H1B
Time Limit: 15 minutes
Status: A solution found.
Nodes Expanded: 63
Time taken: 0.001 seconds
□
```

- It successfully found a solution.
- It has expanded only 63 nodes.
- It found the solution very quickly (0.001 seconds).
- The effect of the use of heuristics is clearly visible.
- It did not have memory problems.

4.1.4 DFS-H2 (Enhanced Warnsdorff) Result

a	b	c	d	e	f	g	h
15	30	11	38	25	28	9	60
12	37	14	29	10	59	24	27
31	16	39	52	45	26	61	8
36	13	46	49	58	53	44	23
17	32	51	40	47	62	7	54
2	35	48	63	50	57	22	43
33	18	1	4	41	20	55	6
0	3	34	19	56	5	42	21

```
erene@DESKTOP-L4U7C0A MINGW64 ~/OneDrive/Masaüstü/ai_project1/ai_pro1/src
```

```
○ $ java -Xmx4g -Xss1g Main 8 DFS-H2 15
Search Method: DFS-H2
Time Limit: 15 minutes
Status: A solution found.
Nodes Expanded: 63
Time taken: 0.001 seconds
```

- It showed the same performance as the DFS-H1B.
- Expanded to 63 nodes
- It found the solution in 0.001 seconds.

4.2 Test results for N=16

4.2.1 BFS (Breadth-First Search) Result

```
erene@DESKTOP-L4U7C0A MINGW64 ~/OneDrive/Masaüstü/ai_project1/ai_pro1/src
⊗ $ java -Xmx4g -Xss1g Main 16 BFS 15
Exception in thread "main" java.lang.OutOfMemoryError: Java heap space
    at State.<init>(State.java:29)
    at TreeSearch.generateSuccessors(TreeSearch.java:272)
    at TreeSearch.search(TreeSearch.java:58)
    at Main.main(Main.java:16)
```

- As in 8x8, it again received a memory failure error (OutOfMemoryError).
- since the state space is much larger for a 16x16 board, BFS's approach to memorizing all levels is completely inadequate.

- This shows that BFS, as the problem size grows, the memory need increases exponentially.

4.2.2 DFS (Depth-First Search) Result

```
erene@DESKTOP-L4U7C0A MINGW64 ~/OneDrive/Masaüstü/ai_project1/ai_pro1/src
❸ $ java -Xmx4g -Xss1g Main 16 DFS 15
Exception in thread "main" java.lang.OutOfMemoryError: Java heap space
    at java.base/java.util.Arrays.copyOf((Arrays.java:3540)
    at java.base/java.lang.AbstractStringBuilder.ensureCapacityInternal(AbstractStringBuilder.java:245)
    at java.base/java.lang.AbstractStringBuilder.append(AbstractStringBuilder.java:590)
    at java.base/java.lang.StringBuilder.append(StringBuilder.java:179)
    at TreeSearch.boardToString(TreeSearch.java:256)
    at TreeSearch.search(TreeSearch.java:60)
    at Main.main(Main.java:16)
```

- DFS also gave a memory error, but at a different point (Arrays.In copyOf and StringBuilder transactions).
- The number of possible situations on the 16x16 board is so large that even DFS's strategy of going deep pushes the memory limits.

4.2.3 DFS-H1B (Warnsdorff's Rule) Result

Solution

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p
35	46	23	40	37	44	21	42	51	70	19	54	67	58	17	56
24	39	36	45	22	41	50	71	20	53	84	69	18	55	66	59
47	34	75	38	49	72	43	52	93	82	125	90	85	68	57	16
76	25	48	73	78	-127	94	81	126	-115	92	83	124	89	60	65
33	74	77	-126	95	80	127	-90	-117	-106	-81	-114	91	86	15	88
26	-125	96	79	-128	-91	-118	-105	-82	-87	-116	-107	-80	123	64	61
97	32	-123	-100	-119	-104	-83	-86	-89	-70	-79	-44	-113	62	87	14
-124	27	-120	-103	-84	-101	-92	-71	-78	-41	-88	-47	-108	-45	122	63
31	98	-99	-122	-93	-76	-85	-40	-69	-48	-43	-32	-19	-112	13	-110
28	-121	30	-75	-102	-67	-72	-77	-42	-33	-12	-49	-46	-109	-22	121
99	-98	105	-94	-73	-56	-39	-68	-11	-50	-31	-20	-15	-18	-111	12
104	29	-74	-63	-66	-59	-10	-51	-34	-13	-16	-7	-30	-21	120	-23
-97	100	-95	106	-57	-64	-55	-38	-5	-8	-29	-14	-17	-2	11	116
2	103	-62	-65	-60	109	-58	-9	-52	-35	-6	-3	-28	117	-24	119
101	-96	1	4	107	-54	111	6	-37	-4	113	8	-1	-26	115	10
0	3	102	-61	110	5	108	-53	112	7	-36	-27	114	9	118	-25

```
erene@DESKTOP-L4U7C0A MINGW64 ~/OneDrive/Masaüstü/ai_project1/ai_pro1/src
❸ $ java -Xmx4g -Xss1g Main 16 DFS-H1B 15
Search Method: DFS-H1B
Time Limit: 15 minutes
Status: A solution found.
Nodes Expanded: 255
Time taken: 0.002 seconds
```

- It successfully found a solution.
- Expanded by 255 nodes (about a 4-fold increase compared to 63 in 8x8).
- It found the solution very quickly again (0.002 seconds).
- Although the board size has increased by 4 times, the loss of performance is minimal.
- It did not have memory problems.

4.2.4 DFS-H2 (Enhanced Warnsdorff) Result

Solution																-	□	×
a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p			
43	46	23	-60	-83	48	21	-116	-85	50	19	-118	55	52	17	122			
24	-61	44	47	22	-57	-84	49	20	-117	-86	51	18	121	56	53			
45	42	-59	-56	-41	-82	-115	-54	-87	-32	-113	126	-119	54	123	16			
-62	25	-42	-81	-58	-55	-8	-23	-114	-53	-88	-33	-112	125	120	57			
41	-80	-63	-40	-11	-24	-17	-52	-7	-22	-31	-110	127	-120	15	124			
26	-43	-78	-25	-64	-9	-12	-15	-18	-51	-6	-89	-34	-111	58	119			
-79	40	-65	-10	-39	-16	-19	-2	-5	-14	-21	-30	-109	-128	-121	14			
80	27	-44	-77	-26	-3	-48	-13	-20	-1	-50	-35	-90	-107	118	59			
39	-76	81	-66	-47	-38	-27	-4	-49	-36	-29	-108	-127	-122	13	116			
28	79	-100	-45	-74	-67	-94	-37	-28	-69	-124	-91	-106	117	60	99			
77	38	-75	82	-99	-46	-73	-68	-95	-92	-105	-126	-123	100	115	12			
34	29	78	-101	-72	91	-98	-93	-70	-125	-96	101	112	107	98	61			
37	76	35	72	83	-102	-71	90	-97	-104	111	108	97	114	11	106			
2	33	30	75	86	73	92	-103	110	89	96	113	102	105	62	65			
31	36	1	4	71	84	87	6	69	94	109	8	67	64	103	10			
0	3	32	85	74	5	70	93	88	7	68	95	104	9	66	63			

```
erene@DESKTOP-L4U7C0A MINGW64 ~/OneDrive/Masaüstü/ai_project1/ai_pro1/src
$ java -Xmx4g -Xss1g Main 16 DFS-H2 15
Search Method: DFS-H2
Time Limit: 15 minutes
Status: A solution found.
Nodes Expanded: 255
Time taken: 0.003 seconds
```

- It showed almost the same performance as the DFS-H1B.
- Expanded to 255 nodes.
- It found the solution in 0.003 seconds.
- There was also no significant performance difference with H1B here.

4.3 Test results for N=32

4.3.1 BFS (Breadth-First Search) Result

```
erene@DESKTOP-L4U7C0A MINGW64 ~/OneDrive/Masaüstü/ai_project1/ai_pro1/src
④ $ java -Xmx4g -Xss1g Main 32 BFS 15
Exception in thread "main" java.lang.OutOfMemoryError: Java heap space
    at java.base/java.util.Arrays.copyOf((Arrays.java:3540)
    at java.base/java.lang.AbstractStringBuilder.ensureCapacityInternal(AbstractStringBuilder.java:245)
    at java.base/java.lang.AbstractStringBuilder.append(AbstractStringBuilder.java:840)
    at java.base/java.lang.StringBuilder.append(StringBuilder.java:253)
    at TreeSearch.boardToString(TreeSearch.java:256)
    at TreeSearch.search(TreeSearch.java:60)
    at Main.main(Main.java:16)
```

- It received a memory failure error again, as in the previous tests.
- The error is Arrays again. Occurred in copyOf and StringBuilder operations.
- For a problem of this size, the memory requirement of the BFS is much higher than the practical limits.

4.3.2 DFS (Depth-First Search) Result

```
erene@DESKTOP-L4U7C0A MINGW64 ~/OneDrive/Masaüstü/ai_project1/ai_pro1/src
④ $ java -Xmx4g -Xss1g Main 32 DFS 15
Exception in thread "main" java.lang.OutOfMemoryError: Java heap space
    at java.base/java.util.Arrays.copyOf((Arrays.java:3540)
    at java.base/java.lang.AbstractStringBuilder.ensureCapacityInternal(AbstractStringBuilder.java:245)
    at java.base/java.lang.AbstractStringBuilder.append(AbstractStringBuilder.java:590)
    at java.base/java.lang.StringBuilder.append(StringBuilder.java:179)
    at TreeSearch.boardToString(TreeSearch.java:256)
    at TreeSearch.search(TreeSearch.java:60)
    at Main.main(Main.java:16)
```

- DFS also encountered the same memory error. The error point is the same as BFS.
- On a 32x32 board, the state space is so large that even a single path-keeping advantage of the DFS is insufficient.

4.3.3 DFS-H1B (Warnsdorff's Rule) Result

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	{	}	~			
75	86	47	80	77	84	45	82	91	96	43	94	-125	110	41	112	-109	-104	39	-106	-77	-72	37	-74	-31	-12	35	-28	-15	-24	33	-26
48	79	76	85	46	81	90	97	44	93	-126	109	42	113	-110	-103	40	-107	-78	-71	38	-75	-32	-11	36	-29	-2	-13	34	-27	-16	-23
87	74	101	78	89	98	83	92	-127	108	95	-124	-111	-102	111	-108	-79	-70	-105	-76	-33	-10	-73	-30	-3	68	7	2	-1	-14	-25	32
102	49	88	99	104	119	-128	107	116	-123	-112	-101	114	-85	-80	-69	-82	-41	-34	-9	10	17	-4	69	8	63	0	75	6	3	-22	-17
73	100	103	120	127	106	117	-122	-113	-100	115	-86	-61	-68	-83	-42	-35	-8	11	18	-5	70	9	62	67	74	79	64	1	76	31	4
50	121	126	105	118	-121	-114	-99	-116	-87	-62	-67	-84	-43	-36	-81	-40	21	-6	71	16	61	122	73	80	65	116	77	90	5	-18	-21
125	72	123	-120	-51	-98	-117	-86	-63	-66	-57	26	-37	-60	-39	22	-7	12	19	60	125	72	81	66	121	78	89	94	117	-20	91	30
122	51	-50	-97	-118	-89	-52	-115	-56	29	-64	-59	-44	27	-126	13	20	59	126	15	82	123	-70	-67	88	-47	120	115	92	95	98	-19
71	124	-119	34	-53	-48	-52	-30	-65	-58	25	28	-123	-38	23	58	-107	14	83	124	-71	-64	87	-48	-69	-66	93	-46	107	118	29	96
52	35	-96	-49	-90	33	44	-47	54	31	-122	-45	24	-125	-108	-127	84	127	-72	-63	86	-49	-68	-65	-22	-45	-36	119	114	97	106	99
-95	70	37	40	43	-54	53	32	-121	-46	55	-124	-109	-94	57	-106	-73	-62	85	-50	-53	40	-21	-44	-37	-32	-23	110	-35	108	113	28
36	53	42	-91	38	45	-120	-113	52	-103	-110	-95	56	-105	-74	-61	-128	-51	-54	-41	-18	-43	-38	5	-20	-1	-34	-31	-24	111	100	105
69	-94	39	46	41	-114	51	-102	-111	-96	-99	-104	-75	-60	-93	-14	-55	10	-17	-52	-39	6	-19	-2	-33	4	-25	-106	109	-30	27	112
54	47	-92	-115	50	-119	-112	-97	-100	-89	-76	-59	-92	-13	-56	11	-16	-5	48	9	-42	-3	95	7	94	-97	0	3	-26	-105	104	101
-93	68	49	-118	-85	16	-101	-88	-77	-98	-91	-12	-57	12	-15	-6	47	64	67	-4	97	8	93	-96	-109	2	-107	-98	-101	102	-29	26
48	55	-84	19	-116	-87	-78	15	-90	-11	-58	13	52	-7	46	63	68	49	98	65	92	-95	110	95	-72	-99	-56	1	-104	-27	-102	103
67	20	-117	-86	17	32	25	-10	27	14	53	-8	45	62	69	50	99	66	91	-94	-111	-52	-73	-34	-55	-108	-71	-100	-57	-66	25	-28
56	-83	18	33	24	-79	28	41	54	-9	44	61	70	51	100	-115	90	-93	-112	-51	-74	-35	-54	25	-70	7	38	71	-68	-103	-58	-65
21	66	23	-80	31	40	55	26	43	60	71	86	101	-116	69	-50	-75	-36	-53	24	-33	8	37	70	-69	6	39	-114	-67	24		
-82	57	34	39	56	29	42	59	72	85	102	-117	88	-91	-114	-49	-76	-37	12	23	-32	9	36	69	26	41	-116	-113	72	5	-64	-59
65	22	-81	30	75	58	73	84	103	-118	87	-90	-41	-48	-77	-38	13	22	-31	10	35	68	27	42	-117	-110	73	40	-115	-112	23	4
58	35	76	57	38	83	104	-119	120	-89	-42	-47	-78	-39	14	21	-30	11	34	67	28	43	-118	-109	74	-97	-100	-111	-86	3	-60	-63
77	64	37	82	105	74	119	-88	-43	-46	-79	-40	15	20	-29	18	33	66	29	44	-119	-108	75	-96	-101	-70	-87	-80	-99	-62	-85	22
36	59	106	79	118	-87	-120	121	-80	-25	-44	51	-28	17	32	65	30	45	-120	-107	76	-95	-102	-71	-88	-79	-98	-39	-84	-81	2	-61
63	78	81	124	-121	122	-81	-24	-45	52	-27	16	49	64	19	46	127	-104	77	-94	-103	-72	-89	-76	-69	-18	-59	-78	-57	-40	21	-82
60	107	62	117	80	-23	-86	53	-28	91	50	63	80	47	124	31	78	-121	-104	-73	-90	-75	-68	-17	-60	-77	-38	-19	-6	-83	-56	1
113	116	125	-122	123	-82	-21	90	59	62	81	48	117	104	79	126	-105	-128	-93	-122	-67	-26	-61	-36	-15	-20	-5	-58	-41	-54	-3	20
108	61	114	-127	-22	-85	54	61	88	99	92	103	82	125	118	123	114	-123	-66	-91	-74	-35	-16	-21	-10	-37	-14	-7	-4	-49	0	-55
115	112	-123	126	-83	-20	89	58	93	60	83	98	105	116	107	-124	-127	-92	113	-34	-27	-62	-25	-30	-13	-8	-11	-42	-53	-2	19	-4
2	109	-128	-119	-126	55	-84	87	84	97	100	95	102	119	122	115	108	-65	-126	-63	112	-31	-28	-9	-22	-43	-46	-1	-48	-3	-50	-1
111	-124	1	4	127	86	-17	6	57	94	-15	8	121	106	-13	10	-125	110	-11	12	-33	-24	-9	14	-29	-12	-7	16	-45	-52	-5	18
0	3	110	-125	-18	5	56	85	-16	7	96	101	-14	9	120	109	-12	11	-64	111	-10	13	-32	-23	-8	15	-44	-47	-6	17	-2	-51

```
erene@DESKTOP-L4U7C0A MINGW64 ~/OneDrive/Masaüstü/ai_project1/ai_pro1/src
```

```
○ $ java -Xmx4g -Xss1g Main 32 DFS-H1B 15
Search Method: DFS-H1B
Time Limit: 15 minutes
Status: A solution found.
Nodes Expanded: 1023
Time taken: 0.003 seconds
□
```

- It successfully found a solution.
- 1023 nodes expanded.
- At 8x8: 63 knots, at 16x16: 255 knots, at 32x32: 1023 knots. This increase is regular and proportional to the size of the board.
- It found the solution very quickly again (0.003 seconds)
- Despite the fact that the board size has increased by 4 times, the performance is still very good.

4.3.4 DFS-H2 (Enhanced Warnsdorff) Result

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	{)	~	□	
-115	-118	47	-122	-107	-110	45	-124	55	72	43	-126	57	110	41	-128	-113	112	39	126	-97	-102	37	124	-83	-68	35	114	37	110	33	112
48	-121	-116	111	46	-123	-108	71	44	-125	56	109	42	-127	-114	111	40	127	-96	-103	38	125	-84	-65	36	123	-82	-69	34	113	106	109
-117	-114	-119	-106	109	70	61	54	73	106	83	58	-115	108	115	-112	-93	-104	113	-98	-85	-32	-101	-48	-67	-64	-61	122	115	108	111	32
-120	49	-112	45	62	53	74	69	82	59	-116	107	115	-111	-92	-105	114	-95	-86	-9	-100	-3	-66	-33	-60	-49	-70	-81	-62	121	116	105
-113	44	-105	52	75	68	81	60	105	84	117	-110	-91	36	-107	-94	-87	-8	-99	0	-5	-10	-31	-2	-47	-34	-63	-58	-71	118	31	120
50	51	46	63	48	87	102	85	118	-117	104	37	-108	-89	44	35	-106	1	14	-7	4	-1	-4	-11	-30	-59	-50	-45	-80	-57	104	117
43	-104	49	76	67	80	95	-128	103	38	-109	-90	43	50	61	-88	45	34	17	2	13	-6	5	-18	-35	-46	-29	-56	-51	-72	119	30
50	51	64	47	88	101	86	119	-118	-127	42	51	60	65	48	-71	62	15	46	33	18	3	12	9	-12	-19	-36	-73	-44	-54	103	
-103	42	77	66	79	96	127	94	39	52	59	66	49	-72	63	66	47	-70	-91	16	11	32	19	6	-17	8	-15	-28	-65	-52	29	-78
52	65	32	89	100	93	120	-119	-126	41	56	-73	64	65	70	-99	-90	67	-96	-59	-86	-51	10	31	20	-13	-20	-37	-74	-43	102	-53
31	-102	41	78	97	126	99	40	53	58	67	64	71	-100	75	68	-97	-92	-87	-58	-95	-68	-85	-52	7	-16	21	-14	-27	-38	-77	28
12	53	90	33	92	121	-120	-125	68	55	-74	57	76	69	-98	-89	-38	-21	-94	-61	-44	-59	-50	-67	30	-53	-64	-21	-22	-75	-42	101
-101	30	13	40	123	98	125	54	-115	70	63	72	-101	74	-37	-22	-93	-88	-39	-46	-57	-62	79	-84	-49	-66	29	-62	-39	-26	27	-76
54	11	-32	91	34	-121	122	69	-124	-75	-114	61	-104	77	-102	-3	-16	-23	-20	-43	-60	-45	-56	-47	-80	-63	-54	-65	-22	23	100	-41
-33	-100	29	14	39	124	-123	-116	71	62	-77	78	73	-36	-11	-24	-7	-2	-15	-40	-47	-42	-63	-78	-83	-48	-45	28	-61	-40	-25	26
-28	55	10	-31	-122	35	72	37	-76	-113	60	-105	80	-103	-6	-13	-4	-17	-8	-19	-64	-53	-66	-55	-46	-81	-34	-55	-44	-23	24	99
-99	-34	-29	28	15	38	-117	104	55	-78	79	52	-35	-12	-25	-10	-1	-14	-27	-50	-41	-48	-77	-82	117	124	-43	-32	27	-60	25	-24
56	-27	-42	9	-30	73	36	-79	-112	51	54	59	-106	81	-32	-5	-26	-9	-18	-29	-52	-65	-54	-67	-76	-33	118	-35	-56	-41	98	25
-43	-98	-35	-10	27	16	103	-118	105	56	-81	82	53	-34	-107	-120	-31	-28	-51	-114	-49	-68	-73	-112	123	116	125	-42	-31	26	-59	24
-52	57	-26	-41	8	-9	74	25	-80	-111	50	87	58	-119	84	-33	-108	-115	-30	-59	-72	-113	-128	-75	126	119	-22	31	-36	-57	-40	97
-97	-44	-51	-36	-11	26	17	102	-119	106	57	-82	83	86	99	-118	-121	-70	-109	-116	-128	-74	-111	122	115	30	9	-20	-23	-30	23	-58
58	53	-76	-25	-40	7	-8	75	24	-83	-110	49	88	-85	-122	85	100	-117	-124	-71	-110	-127	112	127	120	-21	32	-25	4	-37	96	-39
-77	-96	-45	-50	-37	-12	-39	18	101	-120	107	-84	-109	48	91	98	-123	106	101	-126	111	108	121	114	37	10	29	8	-19	-24	-29	22
-94	59	-54	-75	-24	-47	5	-7	76	23	-108	-121	120	89	-86	47	92	97	110	107	102	113	38	43	28	33	36	5	-26	3	-38	95
71	-78	-95	-46	-49	-38	-13	22	19	100	87	108	-107	-122	121	90	-87	46	105	96	109	44	103	34	39	18	11	2	7	-18	21	-28
60	-93	72	-55	-74	-23	-48	5	-6	77	20	99	88	119	-106	-123	122	93	-88	45	104	95	42	27	22	35	6	19	-2	-27	94	-95
73	70	-79	-72	-83	-56	-65	-14	21	4	-5	86	109	98	89	118	-105	-124	123	94	-89	26	23	40	17	20	1	12	-9	-94	-17	20
66	61	-92	75	-80	-73	-22	-57	-64	-15	78	3	-4	85	110	97	90	117	-104	-125	124	41	-90	21	24	15	-8	-1	-12	-3	-96	93
69	74	67	-82	-71	-84	-63	-66	-21	-58	-3	-16	79	2	91	84	111	96	125	115	-103	-126	25	16	-91	0	13	-6	-93	-10	19	-16
2	65	62	-91	76	-81	-70	-87	-62	-57	-20	-59	-2	-17	80	1	92	83	112	95	126	115	-102	-127	14	-7	-92	-11	-4	-13	92	-97
63	68	1	4	-85	78	-89	6	-69	80	-61	8	-19	82	-1	10	81	84	93	12	113	86	127	14	-101	88	-5	16	-99	90	-15	18
0	3	64	77	-90	5	-86	79	-88	7	-68	81	-60	9	-18	83	0	11	82	85	94	13	114	87	-128	15	-100	89	-14	17	-98	91

```
erene@DESKTOP-L4U7C0A MINGW64 ~/OneDrive/Masaüstü/ai_project1/ai_pro1/src
```

```
○ $ java -Xmx4g -Xss1g Main 32 DFS-H2 15
Search Method: DFS-H2
Time Limit: 15 minutes
Status: A solution found.
Nodes Expanded: 1023
Time taken: 0.007 seconds
□
```

- Showed similar performance with DFS-H1B.
- 1023 nodes expanded.
- It found the solution in 0.007 seconds.

- A small performance difference was observed in this dimension compared to H1B.

4.4 Test results for N=41

4.4.1 BFS (Breadth-First Search) Result

```
erene@DESKTOP-L4U7C0A MINGW64 ~/OneDrive/Masaüstü/ai_project1/ai_pro1/src
$ java -Xmx4g -Xss1g Main 41 BFS 15
Exception in thread "main" java.lang.OutOfMemoryError: Java heap space
    at java.base/java.util.Arrays.copyOf((Arrays.java:3540)
    at java.base/java.lang.AbstractStringBuilder.ensureCapacityInternal(AbstractStringBuilder.java:245)
    at java.base/java.lang.AbstractStringBuilder.append(AbstractStringBuilder.java:590)
    at java.base/java.lang.StringBuilder.append(StringBuilder.java:179)
    at TreeSearch.boardToString(TreeSearch.java:256)
    at TreeSearch.search(TreeSearch.java:60)
    at Main.main(Main.java:16)
```

- Again, a memory failure error.
- As with the previous sizes, the BFS is practically unusable in this size as well.

4.4.2 DFS (Depth-First Search) Result

```
erene@DESKTOP-L4U7C0A MINGW64 ~/OneDrive/Masaüstü/ai_project1/ai_pro1/src
$ java -Xmx4g -Xss1g Main 41 DFS 15
Exception in thread "main" java.lang.OutOfMemoryError: Java heap space
    at java.base/java.util.Arrays.copyOfRange((Arrays.java:3852)
    at java.base/java.lang.String.<init>(String.java:4889)
    at java.base/java.lang.StringBuilder.toString(StringBuilder.java:478)
    at TreeSearch.boardToString(TreeSearch.java:259)
    at TreeSearch.search(TreeSearch.java:60)
    at Main.main(Main.java:16)
```

- DFS also gave a memory failure error.
- The advantage of DFS using less memory is insufficient in this dimension.

4.4.3 DFS-H1B (Warnsdorff's Rule) Result

```
erene@DESKTOP-L4U7C0A MINGW64 ~/OneDrive/Masaüstü/ai_project1/ai_pro1/src
$ java -Xmx4g -Xss1g Main 41 DFS-H1B 15
Search Method: DFS-H1B
Time Limit: 15 minutes
Status: Fail - First attempt unsuccessful for board size 41
```

- For the first time we observe the state of failure.
- This result is one of the exceptions mentioned in the article.

4.4.4 DFS-H2 (Enhanced Warnsdorff) Result

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	{		/	~												
109	86	113	120	127	64	123	-122	-109	62	-95	-98	-79	60	-127	78	-77	58	-123	92	-75	56	-119	-73	54	-115	-52	-71	52													
118	115	-128	111	123	-123	-110	125	-120	-107	-88	-93	-96	79	96	-125	90	98	76	-121	74	69	-18	-117	-4	-21	-50	-113	-32	-53												
67	108	117	-123	-111	-102	-119	-106	-81	-92	-99	80	97	-128	89	94	73	-124	91	70	-39	-120	73	68	-17	-116	-5	-22	-49	-112	-33	-54										
116	-127	-116	-103	-118	-105	-82	-91	-100	81	98	127	88	95	74	113	116	71	-40	75	-100	-15	-38	-3	72	67	-16	-31	-6	-23	-48	-57	-34	-55								
107	68	-125	-112	-83	-90	-101	82	99	126	87	98	75	112	117	72	-41	114	-101	-16	-5	-30	-99	-14	-37	-2	71	66	-15	-30	-7	-24	-47	-80	-35	-70	-65	-84	-101	-62	39	
-126	-115	-88	-117	-104	83	100	125	86	97	76	111	118	99	-42	115	-102	-17	-4	-29	40	9	-6	-31	-98	-13	-36	-1	70	65	-14	-29	-8	-25	-46	-67	-82	-77	-95	-91	-102	
69	106	-113	-84	-89	124	85	96	77	110	119	100	-43	-46	-103	-18	-3	-28	41	-80	7	38	67	10	-7	-32	-97	-12	-35	0	69	2	-13	-28	-69	-36	-71	-66	-83	38	-61	
-114	-87	104	123	84	101	78	103	120	101	-44	-47	-104	-19	-2	-27	42	-81	6	39	66	-79	8	37	68	11	-8	-33	-96	-11	64	-9	-26	3	-12	-45	-68	-43	-76	-95	-92	
105	70	-85	102	79	86	95	84	109	-48	-105	-20	-45	-26	43	-82	5	64	-47	-36	-67	-64	69	-78	15	36	17	12	-9	-34	1	68	51	-10	-27	4	-37	-72	-93	-60	37	
-86	103	122	105	110	83	104	121	102	-23	2	-25	44	-1	4	63	-46	-35	-66	65	-48	-37	-68	-63	70	-77	14	38	18	-95	-10	63	36	23	12	-11	-44	5	-42	-75	-94	
71	104	109	80	87	94	88	108	-49	-106	-21	0	3	62	-45	112	-83	-42	-45	-34	-65	-62	-49	10	-69	16	71	-76	13	34	19	52	67	58	35	22	11	-38	-73	-36	59	
108	121	106	111	82	105	122	-107	22	1	-24	45	48	111	-84	-43	-33	-50	-41	-42	11	-38	-61	50	9	-70	-59	72	-75	-94	33	62	37	24	13	26	15	6	-41	-74		
103	72	81	88	93	-114	107	-50	59	46	49	108	61	-44	113	-32	-49	-12	-41	-44	-51	-40	7	12	-53	-60	-81	8	-71	-58	73	20	53	66	49	34	21	10	-39	-58	35	
120	107	112	125	106	123	-108	-115	50	107	60	47	110	-85	-26	-11	-40	-47	2	-1	6	-43	-52	5	-39	10	7	14	-57	-52	5	-74	-93	32	61	38	25	14	27	16	7	-40
73	102	89	92	-113	-116	51	58	-51	116	109	-86	-25	114	-31	-48	-13	-2	71	4	1	68	21	8	13	-54	11	6	15	-72	-55	74	21	54	65	48	33	20	9	34	-57	
90	119	126	113	124	-109	-52	117	106	57	-24	115	-30	-27	-10	39	72	3	0	67	70	5	-122	49	20	9	16	-55	-56	-53	4	-73	-92	31	60	39	56	41	28	17	8	
101	74	91	-112	-117	52	105	56	-23	102	-87	108	9	-38	-29	-14	-3	90	87	-124	-115	66	69	22	121	50	19	12	3	18	75	-54	75	22	55	64	47	32	19	-56	33	
118	127	114	-55	-110	-53	118	101	-88	107	-36	119	-28	109	-4	73	86	109	-114	108	-107	88	123	-121	48	23	14	17	17	21	19	30	-91	24	59	40	57	42	29	18		
75	100	-111	-118	53	104	56	-22	103	116	105	110	-37	-8	-15	-6	91	106	89	-116	-125	-112	63	120	51	122	29	18	13	76	-71	74	23	76	57	10	63	46	31	32	-55	
-128	117	-56	115	-54	119	100	-89	106	111	120	-35	118	-5	74	85	110	-117	108	-113	116	-127	120	40	47	24	15	28	-57	72	1	20	29	-90	25	58	9	58	43	30		
99	76	-119	-66	103	54	-21	94	115	104	117	124	-107	-16	-7	92	105	96	117	-128	-126	119	64	123	52	41	30	17	-70	77	-72	59	26	77	56	11	62	45	-54	31		
116	-127	116	-57	120	-19	90	99	112	121	114	-105	-34	-87	84	75	82	111	-118	97	118	115	124	101	46	39	16	25	-58	27	58	71	0	21	28	-89	38	55	8	59	44	
77	98	-67	-120	-65	102	93	-20	95	98	123	-108	125	-106	-17	88	93	104	95	112	-127	102	63	114	53	42	31	38	-69	78	-73	60	27	56	53	78	7	12	61	30	-53	
-114	115	-126	117	-58	121	-18	-91	122	113	96	127	-104	-33	-86	83	76	81	98	103	78	113	100	43	36	45	-68	-59	-26	77	70	57	22	-1	-88	39	54	37	-2	-13	60	
97	78	-113	-68	-121	-64	101	92	97	-16	-127	90	-109	128	-89	-18	-31	94	77	80	99	62	59	54	-23	32	37	-76	79	-74	61	68	55	24	79	52	15	6	13	-52	9	
114	-115	112	-125	116	-59	122	-17	-92	91	-44	39	-128	-103	-32	-85	-90	-19	-30	61	58	79	-26	33	44	35	-60	-67	-78	69	54	23	-2	-115	40	-87	36	-1	-12	-3	-14	
79	96	119	-111	-65	-122	-63	100	128	98	-15	-126	89	-110	-123	-29	-84	57	-10	45	-102	33	-8	-123	-104	85	-120	-27	-82	47	28	-25	-72	21	-22	-75	-2	-15	14	-13	-24	
83	92	43	56	65	84	93	-108	-121	124	-105	-94	-45	-58	-89	-96	-37	-128	-77	90	-31	-124	-103	86	-121	-28	-81	-68	27	-26	-71	22	-23	76	1	12	19	-10	-23	24		
88	41	52	39	76	57	74	107	94	-107	-124	-59	-88	-97	-36	-125	-78	89	-32	-29	-80	87	26	-65	-70	23	-24	-77	2	11	20	-3	-16	-1	-14	-25	-38					
91	84	89	44	51	66	85	92	73	106	95	90	-123	126	-103	100	-125	-60	-87	-98	-35	-30	-79	88	-33	-64	-69	24	-67	-78	3	10	21	-4	-17	0	-9	-22	-37	-46	23	
2	87	36	47	38	45	50	69	86	91	72	105	96	89	102	-102	-99	-126	-61	-86	-99	-34	-63	-82	25	-66	-79	6	9	22	-5	-18	-29	-8	-21	-32	-43	-26	-39	-36		
85	90	1	4	35	48	67	86	6	33	70	87	8	31	104	97	10	29	-128	-101	12	27	-62	-85	14	25	-80	-83	16	23	4	7	18	-7	20	-31	20	-27	-34	-41	22	-45
0	3	86	37	46	5	34	49	68	7	32	71	88	9	30	103	98	11	28	-127	-100	13	26	-61	-84	15	24	5	6	17	-6	-19	-30	19	-28	-33	-42	21	-44	-35	40	

erene@DESKTOP-L4U7C0A MINGW64 ~/OneDrive/Masaüstü/ai_project1/ai_pro1/src

```
⑧ $ java -Xmx4g -Xss1g Main 41 DFS-H2 15
Search Method: DFS-H2
Time Limit: 15 minutes
Status: A solution found.
Nodes Expanded: 1680
Time taken: 0.01 seconds

● It successfully found a solution.
● 1680 knot expanded.
● At 32x32: 1023 knots,at 41x41: 1680 knots.
● It found the solution in 0.01 seconds.
● Even in the case where H1B failed, it was able to create a solution.
```

4.5 Test results for N=52

4.5.1 BFS (Breadth-First Search) Result

```
erene@DESKTOP-L4U7C0A MINGW64 ~/OneDrive/Masaüstü/ai_project1/ai_pro1/src
⑨ $ java -Xmx4g -Xss1g Main 52 BFS 15
Exception in thread "main" java.lang.OutOfMemoryError: Java heap space
    at java.base/java.util.Arrays.copyOf(Arrays.java:3540)
    at java.base/java.lang.AbstractStringBuilder.ensureCapacityInternal(AbstractStringBuilder.java:245)
    at java.base/java.lang.AbstractStringBuilder.append(AbstractStringBuilder.java:840)
    at java.base/java.lang.StringBuilder.append(StringBuilder.java:253)
    at TreeSearch.boardToString(TreeSearch.java:256)
    at TreeSearch.search(TreeSearch.java:60)
    at Main.main(Main.java:16)
```

With each increase in size, the memory problem of the BFS becomes even more critical.

4.5.2 DFS (Depth-First Search) Result

```
erene@DESKTOP-L4U7C0A MINGW64 ~/OneDrive/Masaüstü/ai_project1/ai_pro1/src
❸ $ java -Xmx4g -Xss1g Main 52 DFS 15
Exception in thread "main" java.lang.OutOfMemoryError: Java heap space
        at java.base/java.util.Arrays.copyOf((Arrays.java:3540)
        at java.base/java.lang.AbstractStringBuilder.ensureCapacityInternal(AbstractStringBuilder.java:245)
        at java.base/java.lang.AbstractStringBuilder.append(AbstractStringBuilder.java:840)
        at java.base/java.lang.StringBuilder.append(StringBuilder.java:253)
        at TreeSearch.boardToString(TreeSearch.java:256)
        at TreeSearch.search(TreeSearch.java:60)
        at Main.main(Main.java:16)
```

- DFS also gave the same memory failure error.
 - Similar error points.

4.5.3 DFS-H1B (Warnsdorff's Rule) Result

```
erene@DESKTOP-L4U7C0A MINGW64 ~/OneDrive/Masaüstü/ai_project1/ai_pro1/src
$ java -Xmx4g -Xss1g Main 52 DFS-H1B 15
Search Method: DFS-H1B
Time Limit: 15 minutes
Status: Fail - First attempt unsuccessful for board size 52
```

- Failed again, as in N=41.
 - This is also one of the special cases mentioned in the article.
 - It was confirmed that N=52 is a problematic dimension for H1B

4.5.4 DFS-H2 (Enhanced Warnsdorff) Results

```
erene@DESKTOP-L4U7C0A MINGW64 ~/OneDrive/Masaüstü/ai_project1/ai_pro1/src
○ $ java -Xmx4g -Xss1g Main 52 DFS-H2 15
Search Method: DFS-H2
Time Limit: 15 minutes
Status: A solution found.
Nodes Expanded: 2703
Time taken: 0.01 seconds
□
```

- It successfully found a solution.
- 2703 nodes expanded.
- At N=41: 1680 nodes, at N=52: 2703 nodes.
- It found the solution again very quickly (0.01 seconds).
- The increase in the number of nodes is proportional to the problem size.

6. Conclusion

In the conducted tests, classical search algorithms BFS and DFS proved impractical for the Knight's Tour problem, encountering memory errors even at the smallest size (8x8). The DFS-H1B algorithm using Warnsdorff's rule showed highly effective performance in small dimensions (63 nodes for N=8, 255 nodes for N=16, 1023 nodes for N=32) and produced solutions in milliseconds, but failed in special cases like N=41 and N=52. The enhanced Warnsdorff algorithm DFS-H2, however, delivered successful results across all tested dimensions, with a reasonable increase in node count (from 63 at N=8 to 2703 at N=52); notably, it could find solutions in just 0.01 seconds even in cases where H1B failed, demonstrating that H2 is the best choice in terms of both reliability and performance.

