Project – 1

Heuristic Search for the White Rook and King versus the Black King Chess Endgame

Brief history of the project

Chess is one of the oldest games that is played even at this generation. Playing the game of chess involves a human to think critically and make well-thought decisions to make the next move and win the game. Due to this very nature of the game of chess, it is very sought after in the field of Artificial Intelligence, where we attempt to make our computer make decisions on its own. This is the true motive of Artificial Intelligence, not just in chess, but any other game or actions like self driving cars, voice recognition. Etc.

The WRKBK is the most elementary chess endgame with three pieces — white king, white rook and the black king. The white and black king can move one position in any direction, and the rook can only move horizontally or vertically for any number of positions. Since white has two pieces and black only has the king, it is expected that white should easily win by enforcing a checkmate by driving the black king to the corner of the chess board, with the rook giving a mate to the black king thereby locking all the squares the black king can move to.

Employed search strategy

The objective of this project was to decide on what moves the white king and rook should make, given that black moves were deterministic. For the white to decide what moves to make, we have to employ a search strategy which searches the chess board, evaluates each position and finally decides which move to make. This process goes on until we achieve a checkmate and the white piece wins.

After researching about all the search strategies discussed in class, I decided to use the **A* Search** because I found it to be easy to understand, and I could relate the algorithm to the chess game very effectively.

Steps:

- Begin at the given starting positions of white pieces for each test cases, and add these positions to the open list. This open list is the list of chess blocks to be considered.
- Examine the list of reachable adjacent blocks from the starting position which are legal chess moves, and add them to the open list and mark it's origin because we will need it to print the path taken to build the solution.
- Remove the starting block from the open list and move it to a closed list since we will be searching the adjacent blocks now.
- We follow the above steps for each of the adjacent blocks we encounter, it is an iterative process.
- There could be multiple adjacent blocks, but the one that we will search is based on the f score, which is the sum of distance between black king and the edge of the chess board (To checkmate with a rook, it's always necessary to have the black king at the edge of the board), the distance between kings and the estimate cost between the pieces by counting the number of blocks.

• Once we select the blocks based on the lowest f score, we drop it from the open list and add it to the closed list. We then look for other adjacent blocks again, that satisfy the legality of chess moves, and that are not already on the closed list.

• We add the obtained blocks to the open list and carry out the above process until our goal state (checkmate) is found. If no such state is found, we conclude that the solution is not available.

Design of the program

In our program, we use the following co-ordinate system as shown:

Image source:

http://tex.stackexchange.com/questions/168281/how-to-draw-a-chessboard-with-numbers

8	(8,1)	(8, 2)	(8,3)	(8, 4)	(8,5)	(8,6)	(8,7)	(8,8)
7	(7,1)	(7, 2)	(7,3)	(7, 4)	(7,5)	(7,6)	(7,7)	(7,8)
6	(6, 1)	(6, 2)	(6,3)	(6, 4)	(6,5)	(6,6)	(6,7)	(6,8)
5	(5, 1)	(5, 2)	(5, 3)	(5, 4)	(5,5)	(5,6)	(5,7)	(5,8)
4	(4,1)	(4, 2)	(4, 3)	(4, 4)	(4,5)	(4,6)	(4,7)	(4,8)
3	(3, 1)	(3,2)	(3,3)	(3, 4)	(3,5)	(3,6)	(3,7)	(3,8)
2		(2,2)						(2,8)
	(2,1)		(2,3)	(2,4)	(2,5)	(2,6)	(2,7)	
1	(1,1) 1	(1,2)	(1,3) 3	(1,4)	(1,5) 5	(1,6)	(1,7) 7	(1,8)

x = row, bottom to top; y = col, left to right

This program was developed in Java.

This program was developed in multiple modules (separate classes). Each of the modules is briefly described:

- **Main.java:** This is the main application file. It manages input, output and runs the test cases. It also runs the search algorithm.
- **Move.java:** This class represents moved made by the chess pieces by the white king and rook and also the black king.

• **SearchEngine.java:** This lass contains implementation of A* search with pruning of losing states for white. Also initializes data structures required by the search strategy.

- **Solution.java:** Solution object to contain a solution to a test case. Printing the number of states visited and moves to the solution.
- **State.java:** Represents the states in a chess board and also allows state manipulation. It also contains state methods used by the search engine like the function to calculate heuristic and methods to generate successors, the legality of chess moves, and deterministically moving the black king.

Results from running the program for the training benchmark

The results are printed in the following format:

- Test Case number
- Number of moves to the solution
- Number of generated states
- Output of the board positions: the starting position, and the position after each white move, using the same format as the input., along with the 6 digit integer that denotes the position of White king, White rook, Black king in the same order. (WKWRBK) = (XXXXXX) representation of co-ordinates. (Please see the design section for representation of co-ordinates).
- All generated states printed, with repeated states included.

Results from Output.txt: (Please check the output file for a better formatted output, also follow instructions in the README document to display output on the console)

TR01
Moves to solution: 3
Generated states: 73
0 (657881):
BK
WK
1 (647882):
BK
WK

All Generated States:

757872 767872 667882 567882 557882 547882 647882 747872 658872 656872 657782 655872 657682 654872 657582 653872 657482 652872 657382 651872 747872 757873 657883 557883 547883 537883 637883 648872 646872 647783 645872 647683 644872 647583 643872 647483 642872 647381 641872 847863 857863 757863 657863 647862 748862 746871 747762 745862 747662 744862 747562 743862 742862 741862 647884 537884 527884 627884 638883 636884 637784 635884 637684 634884 637584 633884 637482 632884 637384 631884 637284 637184

TR02

WK
 2 (C2710E).
2 (637185):
BK
WR
WK
3 (647184):
BK
WR
WK
4 (C 4010 4):
4 (648184):
WR BK
WK

All Generated States:

626876 625876 624876 627586 623876 627486 622876 627386 621876 627286 627186 626576 627485 625576 627385 624576 627885 627285 623576 627185 622576 621576 727175 737175 637185 537185 527185 517185 617185 628175 626175 627285 625175 627385 624175 627485 623175 622175 621175 627885 737175 647184 547184 537184 527184 627184 727175 638175 636175 637284 635175 637384 634175 637486 633175 632175 631175 637784 637884 837165 637165 627165 727165 827165 738165 736176 737265 735165 734165 733165 732165 731165 657185 557185 547185 537185 637185 647485 648184 646185 647285 645185 647385 644185 647485

643185 647583 642185 647685 641185 647785 647885

TR03
Moves to solution: 2
Generated states: 128
0 (657886):
BK
WR
WK
1 (657185):
BK
WR
WK
2 (658185):
WR BK
WK

All Generated States:

667885 567885 557885 547885 647885 747887 658877 656885 655885 657685 654885 657587 653885 657485 652885 657385 651885 657285 657185 666874 566875 556875 546875 646876 657884 655884 656784 658885 654884 656684 653884 652884 651884 667184 567184 557184 547184 647184 658185 656184 657284 655184 657384 654184 657486 653184 657584 652184 657684 651184 657784 657884 665874 565875 555875 545875 645876 656884 654884 655784 655884 655884 655684 655885 652884 655584 651884 655486 655384 655284 655184 667384 567384 557384 547384 647384 658385 656384 657486 657284 655384 657584 657384 657486 657284 655384 657584 657184 654384

657684 653384 657784 652384 657884 651384 661874 561875 551875 541875 641876 652884 651784 653884 651684 654884 651584 655884 651486 656884 651384 657884 651284 658885 651184 667284 567284 557284 547284 647284 658285 656284 657384 657184 655284 657486 654284 657584 653284 657684 652284 657784 651284 657884

TR04 Moves to solution: 4 Generated states: 327 0 (655281): BK -- -- -- ---- -- -- -- -- -- ---- -- -- WK -- -- ---- WR -- -- -- ---- -- -- -- -- ---- -- -- -- -- -- ---- -- -- -- -- -- --__ __ __ __ 1 (745271): __ __ __ __ BK -- -- WK -- -- ---- -- -- -- -- -- ---- WR -- -- -- --__ __ __ __ __ -- -- -- -- -- -- --__ __ __ __ -- -- -- -- -- -- --2 (735261): -- -- -- -- -- -- ---- -- WK -- -- -- --BK -- -- -- -- ---- WR -- -- -- ---- -- -- -- -- ---- -- -- -- -- -- --__ __ __ __ 3 (735571): -- -- -- -- -- -- --BK -- WK -- -- --__ __ __ __ __ -- -- -- WR -- --__ __ __ __ -- -- -- -- -- -- --

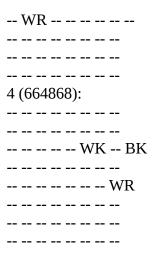
-- -- -- -- -- -- --

4 (735171):
-- -- -- -- -- -- -BK -- WK -- -- -WR -- -- -- -- --- -- -- -- -- --

All Generated States:

845462 745462 645462 635461 736481 734461 735561 735361 737461 733461 735661 735261 738461 732461 735761 735171 731461 735861 835562 845562 745562 645562 635561 736581 734561 735661 735461 737561 733561 735761 735361 738561 732561 735861 735261 731561 735171 835662 845662 745662 645662 635661 736681 734661 735761 735561 737661 733661 735861 735461 738661 732661 735361 731661 735261 735171 631241 641241 541241 441241 431241 532241 531341 531151 533241 531441 534261 531541 535241 531641 536241 531741 537241 531841 538241 **TR05** Moves to solution: 4 Generated states: 408 0 (657867): -- -- -- -- -- ---- -- -- WR -- -- -- WK -- BK --__ __ __ __ -- -- -- -- -- -- ---- -- -- -- -- -- ---- -- -- -- -- -- ---- -- -- -- -- -- --1 (657257): __ __ __ __ __ -- WR -- -- -- ---- -- -- WK -- -- ---- -- BK ---- -- -- -- -- ---- -- -- -- -- -- ---- -- -- -- -- -- ---- -- -- -- -- -- --2 (654258): -- -- -- -- -- -- ---- -- -- -- -- -- ---- -- -- WK -- -- ---- -- -- BK -- WR -- -- -- ---- -- -- -- -- -- ---- -- -- -- -- -- ---- -- -- -- -- -- --3 (664268): __ __ __ __ -- -- -- -- -- -- ---- -- -- WK -- BK

__ __ __ __



All Generated States:

TR06

Moves to solution: 4 Generated states: 756 0 (556473):

-- -- -- -- -- -- ---- -- -- -- -- -- --2 (666185): -- -- BK -- ---- -- -- -- -- -- --WR -- -- -- WK -- ---- -- -- -- -- -- ---- -- -- -- -- -- ---- -- -- -- -- -- ---- -- -- -- -- -- --3 (666486): -- -- BK -- ---- -- -- -- -- -- ---- -- WR -- WK -- ---- -- -- -- -- -- ---- -- -- -- -- -- ---- -- -- -- -- -- ---- -- -- -- -- -- --4 (668486): -- -- WR -- BK -- ---- -- -- -- -- -- ---- -- -- WK -- ---- -- -- -- -- -- ---- -- -- -- -- -- ---- -- -- -- -- -- --

All Generated States:

656472 546472 555463 556574 554463 556674 553463 556774 556174 552463 556874 551463 666573 566573 546573 556675 556475 556775 556875 556275 556175 666764 566775 466775 456775 446775 546775 557763 555775 556875 556675 558775 554775 556573 553775 556475 552775 551775 556275 556175 666185 566175 466175 456175 446175 546175 557163 555175 556275 558175 554175 553175 556475 552175 556573 551175 556675 556775 556875 666864 566875 466875 456875 446875 546875 557863 555875 556775 558875 554875 556675 553875 556573 552875 556475 551875 556275 556175 776175 676175 576175 566175 556175 656184 667184 665174 666274 668174 664174 666374 663174 666486 662174 666574 661174 566675 466675 456675 446675 546675 557663 555675 556775 556573 558675 554675 556875 556475 553675 552675 556275 551675

TR07
Moves to solution: 4
Generated states: 501
0 (556476):
BK
WR
WK
1 (556275):
BK
WR
WK
2 (557285):
BK
WR
WK
3 (647284):
BK
WR WK
VV IX
4 (648284):
WR BK
WK

-- -- -- -- -- -- --

All Generated States:

TR08

-- -- -- -- -- -- --

Moves to solution: 4

-- -- -- -- -- -- ---- -- -- -- -- --2 (537461): -- -- -- -- -- -- ---- -- WR -- -- --BK -- -- -- -- ---- -- WK -- -- -- ---- -- -- -- -- -- ---- -- -- -- -- -- ---- -- -- -- -- -- --3 (537551): -- -- -- -- -- -- ---- -- -- WR -- -- ---- -- -- -- -- -- --BK -- WK -- -- ---- -- -- -- -- -- ---- -- -- -- -- -- --4 (537151): -- -- -- -- -- -- --WR -- -- -- -- --__ __ __ __ BK -- WK -- -- ---- -- -- -- -- -- ---- -- -- -- -- -- ---- -- -- -- -- -- --

UHID: 1430872

Date: 10/06/2016

Name: Adithya Srinivasa

All Generated States:

546472 535472 536574 536374 534472 536674 536274 533472 536774 536174 532472 536874 531472 645462 445463 435463 425463 525473 536473 534473 535573 537461 533473 535673 538473 532473 535773 531473 535873 641462 541462 441463 431463 421463 521473 532473 531573 531373 533473 531673 531273 534473 531773 531173 535473 531873 536473 537461 538473 546375 526375 626375 536475 536275 536175 536675 536775 536875 643462 543462 443463 433463 423463 523473 534473 532473 533573 533373 535473 531473 533673 533273 536473 533773 533173 537461 533873 538473 546775 446775 436775 426775 526775 626775 537765 535765 536875 536675 538765 534765 533765 536475 532765 536375 531765 536275 536175 644462 544462 434463 424463 524473 535473 533473 534573

TR09

Moves to solution: 4 Generated states: 146

0 (434563):

WK WR
1 (424462).
1 (434462):
BK
WK WR
2 (436451):
WR
BK WK
3 (436541):
WR
BK WK
4 (436141):
WR
BK WK
All Generated States: 444564 344554 334554 324554 424554 435564 433564
434664 434462 436574 432564 434764 437564 431564
434864 438564 544452 344453 334453 324453 424463
435463 433463 434563 436451 432463 434663 437463

431463 434763 438463 434863 536441 546452 446452

346452 336452 326452 437462 435462 436541 436341 438462 434462 436641 433462 436741 432462 436841 431462 354554 344554 334554 434563 445563 443563 444665 442563 444765 441563 444865 448563 644451 654453 554453 454453 543451 544551 544351 542451 544651 541451 544751 544851 536551 546552 446552 346552 336552 437551 435531 436651 436451 438551 434551 436751 436351 433551 436851 436251 432551 436141 431551 536351 546352 446352 346352 336352 437351 435331 436451 436251 438351 436551 436141 436651 436751 436851 536651 546652 446652 346652 336652 437651 435631 436751 436551 438651 434651 436851 436451 433651 436351 432651 436251 431651 436141 536751 546752 446752 346752 336752 437751 435731 436851 436651 438751 434751 436551 433751 436451 432751 436351 431751 436251 436141

TR10

Moves to solution: 4 Generated states: 253 0 (441381):

-- -- WK -- -- --

-- -- -- WR -- -- -- --1 (531372):

-- -- WK -- -- -- --

-- -- -- -- -- -- --

-- -- WR -- -- -- --

2 (521371):

BK -- -- -- -- -- -- -- -- -- --

-- WK -- -- -- -- --

-- -- -- -- -- -- ---- -- WR -- -- -- --

All Generated States:

421864 521864 532874 531774 533874 531674 534874 531574 535874 531472 536874 531374 537884 531274 538874 531174 731352 741352 641352 541352 531351 632351 631451 631251 633351 631551 631161 634351 631651 635371 631751 631851 431363 331353 321352 311352 411361 422361 421463 421263 423361 421563 421163 424361 421663 425361 421763 421863 428361 542374 442364 432364 422364 522364 533374 531374 532472 532274 534374 532574 532174 532674 532774 532874

TR11

Moves to solution: 4 Generated states: 119 0 (433363):

-- -- -- -- -- --

-- -- BK -- -- -- ---- -- -- BK

-- -- WK -- -- -- ---- -- WR -- -- -- --

-- -- -- -- -- -- -- --

1 (433462):

-- BK -- -- -- -- ---- -- -- -- -- -- --

-- -- WK -- -- -- ---- -- WR -- -- --

-- -- -- -- -- -- --

2 (436451): -- -- -- -- -- -- --

-- -- -- WR -- -- --

BK -- -- -- -- -- -- -- -- -- -- WK -- -- -- --

-- -- -- -- -- -- --

3 (436541):

-- -- -- -- -- -- --

-- -- -- WR -- -- --

BK -- WK -- -- --

Name: Adithya Srinivasa UHID: 1430872 -- -- -- -- -- -- ---- -- -- -- -- --4 (436141): -- -- -- -- -- -- ---- -- -- -- -- -- --WR -- -- -- -- ---- -- -- -- -- -- --BK -- WK -- -- ---- -- -- -- -- -- ---- -- -- -- -- -- ---- -- -- -- -- -- --All Generated States: 443364 343354 323354 423354 432364 433462 433264 431364 433564 433164 433664 433764 433864 543452 443452 333453 323453 423463 434463 432463 433563 433363 435463 431463 433663 433263 436451 433763 433163 437463 433863 438463 536441 546452 446452 346452 336452 326452 437462 435462 436541 436341 438462 434462 436641 433462 436741 432462 436841 431462 453365 353355 343355 433355 444365 442365 443465 443265 445365 441365 443563 443165 443665 443765 448365 443865 536551 546552 446552 346552 336552 437551 435531 436651 436451 438551 434551 436751 436351 433551 436851 436251 432551 436141 431551 536351 546352 446352 346352 336352 437351 435331 436451 436251 438351 436551 436141 436651 436751 436851 536651 546652 446652 346652 336652 437651 435631 436751 436551 438651 434651 436851 436451 433651 436351 432651 436251 431651 436141 **TR12** Moves to solution: 6 Generated states: 3455 0 (828184): WR WK -- BK -- -- ---- -- -- -- -- -- ---- -- -- -- -- -- ---- -- -- -- -- -- --__ __ __ __ -- -- -- -- -- -- --1 (826175): -- WK -- -- -- --

Date: 10/06/2016

Name: Adithya Srinivasa -- -- BK -- --WR -- -- -- -- --__ __ __ __ __ -- -- -- -- -- -- ---- -- -- -- -- -- ---- -- -- -- -- -- ---- -- -- -- -- --2 (836176): -- -- WK -- -- ---- -- BK -- --WR -- -- -- ---- -- -- -- -- -- ---- -- -- -- -- -- ---- -- -- -- -- -- ---- -- -- -- -- -- ---- -- -- -- -- -- --3 (746177): -- -- -- -- -- ---- -- WK -- -- BK --WR -- -- -- -- ---- -- -- -- -- -- ---- -- -- -- -- -- ---- -- -- -- -- -- ---- -- -- -- -- -- ---- -- -- -- -- -- --4 (756178): -- -- -- -- -- ---- -- -- WK -- -- BK WR -- -- -- -- ---- -- -- -- -- -- ---- -- -- -- -- -- ---- -- -- -- -- -- ---- -- -- -- -- -- --__ __ __ __ __ 5 (866188): -- -- -- WK -- BK -- -- -- -- -- -- --WR -- -- -- -- --

UHID: 1430872

Date: 10/06/2016

-- -- -- -- -- -- ---- -- -- -- -- -- ---- -- -- -- -- ---- -- -- -- -- -- --6 (866888): -- -- WK -- BK -- -- -- -- -- -- ---- -- -- WR

All Generated States:

TR13

-- -- -- -- -- -- --

Name: Adithya Srinivasa UHID: 1430872 -- -- -- -- -- -- ---- -- WK -- -- -- --__ __ __ __ -- BK -- -- -- ---- -- -- -- -- -- ---- -- -- -- -- -- --2 (533441): -- -- -- -- -- -- ---- -- -- -- -- -- ---- -- -- -- -- -- ---- -- WK -- -- --BK -- -- -- -- ---- -- WR -- -- ---- -- -- -- -- -- ---- -- -- -- -- -- --3 (533351): -- -- -- -- -- -- --

Date: 10/06/2016

4 (533151):

WR -- -- -- -- -- -- --

All Generated States:

748444 758443 658443 558443 548434 538432 638443 738443 647444 648544 648344 648644 648244 648744 648144 648844 638443 648443 548442 448442 528433 628443 537433 538533 538333 536433 538633 538233 535433 538733 538133 534433 538833 533441 532433 531433 633451 643452 543452 443452 433451 623442 534451 532451 533551 533351 535451 531451 533651 536451 533751 537451 533851 538451 648445 658445 558435 548235 548735 548135 548835 548452 558453 458453 358453 348453 447452 448552 448852 634461

```
644452 544452 434462 535441 533441 534561 534361 536441 532441 534661 534261 537441 531441 534761 538441 534861 632442 642452 542452 442452 432462 533441 531441 532541 532341 534461 532641 532241 535441 532741 532151 536441 532841 537441 538441 633542 643552 543552 443552 433562 534561 532541 533641 533441 535541 531541 533741 533341 536541 533841 533241 537541 533151 538541 633342 643352 543352 443352 433362 534361 532341 533441 533241 5331341 533541 533151 533641 533741 533841 635442 645442 445442 435462 536441 534461 535541 537441 533441 535641 538441 535441 535441 535441 531341 5333441 531641 531241 534461 531741 531151 535441 531841 531841 536441 537441 538441
```

References:

- http://www.policyalmanac.org/games/aStarTutorial.htm
- https://chessprogramming.wikispaces.com/KRK
- Brief discussion with fellow students about which heuristic will suit the best for this problem.