Name: Nguyen Huu Khang

ID: 18125086

LAB 01 REPORT:

Task completion:

Run:

warnsdoff.py -px 0 -py 0 -s 8

backtracking.py -px 0 -py 0 -s 5

1. **Manipulate the input and output**
2. **Implement helper function (utils.py)**

+ is_safe: check whether a move is possible

+ print_solution: print the tour after found it

+ is_closed_tour: check if a tour is closed

3. **Implement the backtracking (backtracking.py)**

Because running for board size >= 8 is not possible so I don't run the algorithm for board size >= 8.

4. **Implement the Warnsdorff's Heuristic (warnsdoff.py)**

If board size is odd number: The algorithm run only once to find an open tour.

Or else the algorithm constantly finds the path and check to get a closed tour. The limit is 100 runs.

Tie in degree is solved by randomly choose the orders of squares to visit.

5. **Provide valid results for the backtracking strategy:**

```
-px 1 -py 1 -s 5
---Time run--- 0.6306171417236328 (ms)
23 16 11 6 21
10 1 22 17 12
15 24 5 20 7
2 9 18 13 4
25 14 3 8 19
-px 1 -py 1 -s 8
---Time run--- very large
Can not get the result in any board >8 by using backtracking
```

Find an open tour solution for 5x5, can't run in 8x8 board.

6. **Provide valid results for the Warnsdorff's Heuristic.**

```
-px 1 -py 1 -s 5
--- Time run --- 3013 (ms)
Open tour solution:
21 6 17 12 19
16 1 20 5 10
7 22 11 18 13
2 15 24 9 4
23 8 3 14 25

-px 1 -py 1 -s 8
Find closed tour after 14 runs
--- Time run --- 27356 (ms)
39 44 15 20 35 24 13 22
16 1 38 43 14 21 36 25
45 40 19 34 37 60 23 12
2 17 64 59 42 33 26 51
63 46 41 18 61 50 11 32
6 3 62 55 58 29 52 27
47 56 5 8 49 54 31 10
4 7 48 57 30 9 28 53

-px 1 -py 1 -s 15
--- Time run --- 3041 (ms)
Open tour solution:
69 60 27 2 71 62 29 4 31 86 93 6 33 36 91
26 1 70 61 28 3 102 85 94 5 32 87 92 7 34
59 68 25 72 103 84 63 30 135 170 95 142 35 90 37
24 73 66 83 64 101 134 169 96 147 136 175 88 143 8
67 58 75 104 81 166 97 148 201 176 171 146 141 38 89
74 23 82 65 100 133 206 209 168 149 202 137 174 9 144
57 76 105 80 165 98 167 200 207 210 177 172 145 140 39
22 79 56 99 132 205 208 213 198 203 150 211 138 173 10
55 106 77 164 123 162 199 204 219 212 217 178 181 40 139
78 21 112 131 114 195 220 197 214 189 224 151 216 11 180
107 54 115 122 163 124 161 194 225 218 215 190 179 182 41
20 111 126 113 130 121 196 221 158 193 188 223 152 191 12
53 108 51 116 125 160 129 120 187 222 157 192 183 42 153
50 19 110 127 48 17 118 159 46 15 186 155 44 13 184
109 52 49 18 117 128 47 16 119 156 45 14 185 154 43
```

```
-px 1 -py 1 -s 25
--- Time run --- 3098 (ms)
Open tour solution:
51 46 67 64 53 44 55 474 59 42 57 470 465 40 379 382 393 38 377 362 385 36 367 360 365
66 1 52 45 68 477 60 43 56 481 468 41 380 537 464 39 378 383 392 37 376 361 364 35 368
47 50 65 256 63 54 475 484 473 58 471 540 469 466 381 538 391 394 401 384 363 386 375 366 359
2 257 48 69 476 489 478 61 480 557 482 467 546 539 536 463 410 455 390 395 402 373 358 369 34
49 70 255 488 259 62 485 558 483 472 549 556 541 532 545 456 535 406 409 400 389 396 387 374 357
254 3 258 237 486 495 490 479 616 555 560 547 550 553 542 533 462 411 454 405 408 403 372 33 370
71 238 487 252 491 260 621 494 559 548 617 554 563 544 531 552 457 534 407 412 399 388 397 356 329
4 253 236 261 496 493 500 611 618 615 564 561 598 551 566 543 530 461 416 453 404 413 328 371 32
235 72 239 492 251 622 497 620 501 610 597 614 565 562 575 458 567 452 529 420 415 398 355 330 327
230 5 250 233 262 499 502 625 612 619 606 601 574 599 582 569 528 421 460 417 450 331 414 31 334
73 234 231 240 503 248 623 498 609 590 613 596 605 580 573 576 459 568 451 422 419 354 333 326 143
6 229 214 249 232 263 504 515 624 607 602 589 600 583 594 581 570 527 418 449 332 423 144 335 30
213 74 207 228 241 508 247 608 505 516 591 604 595 572 579 526 577 448 523 426 353 338 325 142 145
116 7 212 215 206 227 264 507 514 603 588 517 584 593 586 571 524 427 352 339 424 147 336 29 152
75 208 117 204 211 242 509 246 437 506 511 592 587 518 525 578 447 522 425 350 337 324 153 146 141
8 115 102 209 216 205 226 265 510 513 440 519 432 585 446 521 428 351 340 319 148 159 140 151 28
101 76 203 118 187 210 243 438 245 436 431 512 441 520 429 346 445 318 349 158 323 294 149 154 139
112 9 114 103 202 217 188 225 266 439 442 433 430 345 444 317 348 341 320 293 310 157 160 27 150
77 100 111 186 119 108 201 244 435 224 269 344 443 314 347 342 297 312 309 322 295 280 291 138 155
10 113 104 97 110 185 218 189 200 267 434 223 270 343 316 313 308 321 296 311 292 305 156 161 26
99 78 95 120 107 122 109 180 219 192 199 268 315 222 301 274 285 298 307 304 281 290 279 168 137
82 11 98 105 96 125 184 123 190 179 220 193 198 271 286 299 302 273 284 289 306 169 282 25 162
79 90 81 94 121 106 87 126 181 194 191 178 221 300 197 272 275 288 303 170 283 278 165 136 167
12 83 92 89 14 85 124 183 16 177 128 195 18 175 130 287 20 173 132 277 22 171 134 163 24
91 80 13 84 93 88 15 86 127 182 17 176 129 196 19 174 131 276 21 172 133 164 23 166 135
```

```
-px 1 -py 1 -s 40
After 1000 runs, find open tour solution but can not find any closed tour.
--- Time run --- 3252 (ms)
```

(The answer is too long so I don't post it here)

- Find open tour solution for all board size: 5x5, 8x8, 15x15, 25x25, 40x40
- Find a closed tour solution for 8x8 board
- Can't find closed tour solution for 40x40

7. **Provide all evidential files in the OUTPUT folder (18125086_backtracking.txt and 18125086_warnsdoff.txt)**

References

[1] https://en.wikipedia.org/wiki/Knight%27s_tour

[2] https://www.geeksforgeeks.org/the-knights-tour-problem-backtracking-1/

[3] https://www.geeksforgeeks.org/warnsdorffs-algorithm-knights-tour-problem/