# 2. Seminarska naloga

Seminarsko nalogo sva napisala v Javi.  
Skladisce sva predstavila v razredu Warehouse. Skladisce hrani podatke o visini in sirini skladisca, trenutnem stanju (*char[][]*), koncnem stanju (*char[][]*), seznam narejenih premikov in oceno trenutnega stanja, glede na koncno stanje.

Vozlisce predstavlja vsako unikatno stanje skladisca. Povezave med vozlisci so vsi veljavni premiki trenutnega stanja.

Oceno trenutnega stanja izracunava po vsakem premiku. Oceno izracunava kot razdaljo polozaja posamezne “skatle” v trenutnem stanju do polozaja v koncnem stanju. Idealna ocena je 0, kjer so vse skatle na pravem mestu. Ce je skatla v pravem stolpcu a v napacni vrstici, oceno kaznujeva s pristevanjem dodatne vrednosti (stevilo stoplcev), saj mora, da skatlo postavi na pravo mesto narediti se vsaj 3 premike.

## BFS

**Primer 1:**

MOVES: 5

Explored nodes: 112

Max depth: 5

Max memory: 90 warehouses

Time in seconds: 0.012

**Primer 2:**

MOVES: 9

Explored nodes: 3006

Max depth: 9

Max memory: 1728 warehouses

Time in seconds: 0.117

**Primer 3:**

MOVES: 8

Explored nodes: 289057

Max depth: 8

Max memory: 472481 warehouses

Time in seconds: 6.632

**Primer 4:**

MOVES: 12

Explored nodes: 3807583

Max depth: 12

Max memory: 1238690 warehouses

Time in seconds: 38.863

## DFS

Primer 1:

MOVES: 36

Explored nodes: 109

Max depth: 41

Max memory: 53 warehouses

Time in seconds: 0.008

Primer 2:

MOVES: 623

Explored nodes: 2877

Max depth: 1059

Max memory: 1379 warehouses

Time in seconds: 0.274

Primer 3:

## A\*

**Primer 1:**

MOVES: 5

Explored nodes: 26

Max depth: 5

Max memory: 34 warehouses

Time in seconds: 0.008

**Primer 2:**

MOVES: 9

Explored nodes: 548

Max depth: 9

Max memory: 574 warehouses

Time in seconds: 0.036

**Primer 3:**

MOVES: 8

Explored nodes: 1444

Max depth: 8

Max memory: 7672 warehouses

Time in seconds: 0.112

**Primer 4:**

MOVES: 12

Explored nodes: 37054

Max depth: 12

Max memory: 129971 warehouses

Time in seconds: 0.984

**Primer 5:**

MOVES: 8

Explored nodes: 2073

Max depth: 8

Max memory: 15891 warehouses

Time in seconds: 0.221

## Genetski algoritem

**Primer 1:**

MOVES: 5

Explored nodes: 250350

Max depth: 9

Max memory: 50 warehouses

Time in seconds: 6.785

**Primer 2:**

MOVES: 9

Explored nodes: 455700

Max depth: 12

Max memory: 50 warehouses

Time in seconds: 10.595

**Primer 3:**

MOVES: ~10

Explored nodes: 539850

Max depth: 20

Max memory: 50 warehouses

Time in seconds: 9.069

**Primer 4:**

MOVES: ~12

Explored nodes: 705300

Max depth: 25

Max memory: 50 warehouses

Time in seconds: 19.472

**Primer 5:**

MOVES: ~11

Explored nodes: 661900

Max depth: 999

Max memory: 50 warehouses

Time in seconds: 13.515

## ID search

**Primer 1:**

MOVES: 7

Explored nodes: 121

Max depth: 7

Max memory: 40 warehouses

Time in seconds: 0.022

**Primer 3:**

MOVES: 15

Explored nodes: 29184

Max depth: 15

Max memory: 5983 warehouses

Time in seconds: 0.426

**Primer 5:**

MOVES: 12

Explored nodes: 34485

Max depth: 12

Max memory: 8028 warehouses

Time in seconds: 0.787

**Primer 2:**

MOVES: 10

Explored nodes: 819

Max depth: 10

Max memory: 241 warehouses

Time in seconds: 0.036

**Primer 4:**

MOVES: 17

Explored nodes: 67597

Max depth: 17

Max memory: 13370 warehouses

Time in seconds: 0.966

## Rezultati

Implementirala sva različne algoritme (BFS, DFS, A\*, Genetski, IDDFS) za reševanje problema skladišča v Javi in merila njihovo učinkovitost z več testnimi primeri.

BFS in A\* sta popolna in optimalna algoritma, ki vedno najdeta optimalno rešitev, če ta obstaja. Čeprav je BFS v manjših testnih primerih po učinkovitosti blizu A\*, vendar pa lahko v večjih primerih porabi precej več spomina in traja dlje časa. Enako velja za DFS, tako da večjih nalog za ta dva algoritma sploh nisva upoštevala.

DFS, IDDFS in Genetski algoritem niso popolni in v nekaterih primerih ne najdejo optimalne rešitve.

Za naše primere so se pri najmanjših nalogah (prvi in drugi primer) dobro izkazali vsi algoritmi, razen genetskega, pri večjih pa sta najboljše rezultate dobila IDDFS in A\*.