

## Hill Climbing

Lõin kaks objekti, Board ja Queen. Board hoiab endas maatriksit nimega *matrix* seisu printimiseks ja listi *queens* Queen objektidega. Board klassist saab teha instancesse, kus on arvutatud laua seis (*value* = kui palju konflikte on).

Alguses loome initial board'i, kus queen'id on paigutatud igaüks oma veerule ning 0 indeksiga reale.

Käivitame tsükli, mis arvutab vastavalt  $N$ 'ile kõikide queen'i paigutustega (queen'i liiguvad ainult mööda oma veergu, ehk  $N - 1$  käiku iga queen'i kohta) *board*'id *value*'dega, mis lisatakse listi *boards*.

Sellest *boards* listist antakse funktsioonile *hill\_climbing(board)* kaasa board, mille value on kõige väiksem (ehk kõige vähesemate konfliktidega board).

Edasi *hill\_climbing* funktsioon võrdleb, kas antud board'i value on väiksem kui eelmine parim, kui jah selle uueks parimaks. Kui board'i value on 0, siis peatakse otsing ning lahendus on leitud, kui mitte siis jätkakse. Kui 0 väärtust ei saavutata pärast  $N^N$ , tagastatakse senini parim board. Üle 6 suuruse laua peal see algoritm enam ei sobi.

Debuggida ei jõudnud, kuid lahenduskäik peaks viima tulemuseni.