

Selle ülesanne eesmärgiks oli paigutada lippusid niimodi, et need ei olnud samas reas, veerus ning diagonaalis. Programm koosneb kahest osast: Hill climbing algoritmist ja NQPosition klassist, kus realiseeritud board-i loogika (kus asuvad lipud).

Alguseks, NQPosition klassi loomisel kasutades randomi ma paigutan lippusid lauda. Laud esitab sellest ühemõõtmelist massiivi (list), kus iga number näitab lipu asukoht y-teljel ja listi index on x-telg.

Hill climbing algoritm otsib optimaalset lahendust niimodi: igas iteratsioonis peab olema leitud väärtus (value), mis on parem kui eelmine. Vastasel juhul ta lõpeb. Selles ülesannes value tähendab konfliktide arvu: on vaja kontrollida read ja diagonaalid. Otsime konfliktide arvu (value), järgmine valem abistab leida konflikte diagonaalides.

- Samm1: 2 diagonaali määramine võrranditega:
 - $X = Y + X_0 - Y_0$
 - $X = -Y + (X_0 + Y_0)$mis lõikuvad positsioonis (X_0, Y_0) .

Joonis 1 Valem diagonaalide väärtuste leidmiseks[1]

Best move meetodis on vaja leida parim move ja value hetkel momendil. Ma genereerin sammu kasutades randomi. Pärast ma kontrollin, kas see samm pareneb value. Kui jah, siis tagastan, vastasel juhul genereerin uuesti tsüklis.

[1] [Loogiline Programmeerimine. Kabe](#)