

# Ülesanne 1: Räsimine

Räsimine tähendab ühesuunalist tehet, kus samast sisendist moodustub alati ühesugune räsi, kuid teistpidi tegevus on võimatu – räsist pole võimalik tuletada algset sõnumit.

Räsimine aitab hoida asju korras ja turvalisena. Kuna iga andmetükk annab ainulaadse räsi, on lihtne kontrollida, kas andmed on õiged või mitte. Kui midagi on muutunud, siis muutub ka räsi väärtus, nii et saab kohe aru, kui midagi on valesti.

Räsimit kasutatakse ka turvalisuse tagamiseks. Näiteks kui salvestame paroole, ei soovi me neid tekstina salvestada, kus igaüks neid lugeda võiks. Selle asemel räsime parooli ja salvestame räsi väärtuse. Kui keegi sisestab parooli, räsitakse sisestatud parooli samamoodi ja vaadatakse, kas see klappib salvestatud räsi väärtusega. Kui see klappib, siis oli parool õige. Kui ei klapi, siis oli parool vale. Ja kuna räsi on ühesuunaline, ei saa keegi räsi väärtusest algset parooli välja nuputada.

## Hea Räsifunktsiooni omadused

### Kiire arvutamine

*Räsi väärtuse arvutamine peaks olema kiire igale antud sisendile.*

### Deterministlikkus

*Sama sisendi korral peab alati tulemuseks olema sama räsi väärtus.*

### Kokkupuõrkekindlus

*Erinevate sisendite puhul ei tohiks tekkida sama räsi väärtust.*

### Mitte-pööratavus

*Algsisendi taastamine räsi väärtuse põhjal peaks olema keeruline.*

# Kokkupõrgete lahendamise tehnikad

## Eraldi aheldamine

Eraldi aheldamine on üks levinumaid viise kokkupõrgete lahendamiseks.

Räsitabelis loome iga indeksi juurde nimekirja ahela, kus saame hoida kõiki sama indeksiga võtmeid. Kui uus võti genereerib räsi väärtuse, mis on juba olemas, lisame selle võtme olemasoleva indeksi ahelasse. See võimaldab meil hoida mitut elementi samas räsitabeli asukohas, eraldades need lingitud nimekirjaga.

## Avatud aadressimine

Avatud aadressimine sarnaneb mängule, kus tuleb leida tühi koht. Kui meie esmane indeks on hõivatud, hakkame tabelis ringi liikuma, kasutades kindlat jada või komputatsiooni, et leida uus tühi koht.

Sõltuvalt sondeerimise meetodist võime liikuda edasi järgmisse lahtrisse, hüpata üle mitme lahtri või järgida mõnda keerulisemat mustrit. Kui leitakse tühi koht, salvestatakse uus element sinna.

See meetod nõuab hoolikat planeerimist, et vältida liiga paljude elementide kogunemist ühte piirkonda, mis võib viia jõudluse languseni.