

Praktikum nr 3

Eesmärk:

Õppida kasutama andmestruktuuri "list" ja lihtsamaid võtteid rekursiooni programmeerimisel.

Ülesanne

1. Tutvuda listi ja rekursiooni mõistega näite "Marsruutide planeerimine" põhjal (vt linki "Näidismaterjal").

2. Laiendada lennuühenduste kohta käivat faktide "lend/2" hulka.

Näide: `lend('Moskva', 'Tallinn', 3000).`

Fakti mall: `lend(lähtekoht, sihtkoht, kaugus/hind).`

3. Programmeerida kitsendusteta otsingureegel marsruudi leidmiseks kahe punkti vahel kasutades selleks rekursiooni.

4. Lisada marsruudi otsingureeglile kitsendus, et juba külastatud linna uuesti ei läbita.

Selleks

- lisada reegli päisesse parameter, mis väärtustatakse läbitud linnade listiga ja
- elemendi sisalduvuse kontrollimiseks listis kasutada süsteemi predikaati `member(element, hulk).`

5. Täiendada otsingureeglit väljumisaja kitsendusega

- Kellaeg päringus näitab varaseimat soovitud väljumisaega.
- Kellaeg esitada listi kujul [tunnid, minutid].
- Täiendada fakte `lend/3` ja otsingureeglit nii, et leitakse ajaliselt lühim marsruut
- Aja ja teepikkuste summeerimiseks kasutada süsteemi operaatorit "is". Näiteks avaldis "X is Y + Z." arvutab muutujale X väärtuse, mis on muutujate Y ja Z väärtuste summa.
- Lennuühenduste ajalise info esitamiseks kasutada fakti malle `lend/5`, kus

`lend(Place_Departure, Place_Arrival, Departure_time, Arrival_time, Duration).`

Kitsendus: Lendu saab jätkata transiitpunktist ainult niisuguse lennuga, mis väljub mitte varem kui 1 tund peale sellesse punkti saabumist.

Märkus

Soovitav on koostada abireeglid teheteks kellaegadega: (kellaegade liitmine ja lahutamine). Reisi kestvuse arvutamisel tuleb arvesse võtta ka mitmepäevaseid marsruute.