

Algoritmika – arvutiteaduse üks alustalasid.

<http://courses.cs.ut.ee/2009/algorithmics/> Arvutiteaduse instituut: www.cs.ut.ee

Algoritm – eeskiri, samm-sammuline juhend, mida järgides saab lahendada ülesandeid.

Algoritm on välja mõeldud inimeste poolt ja seda esitatakse inimestele.

Algoritmi esitamiseks saab kasutada erinevaid esitusviise – „madalal tasemel“ (programmeerimiskeele sarnaselt) või „kõrgemal tasemel“ – piisavalt täpse vaba tekstina.

Arvutiprogramm, mille kirjutab programmeerija, realiseerib algoritme arvutile arusaadavas „keeles“.

Kokaraamatu retsept on hea näide algoritmist: pannakse kirja sisendid (toorained) ja töötlemisjuhised. Loodetavasti saab pärast ka soovitud tulemuse (maitsev toit) .

Arvutiteaduses on algoritmidel keskne koht – oluline on osata hinnata kui kiire on üks või teine meetod. Sest sama ülesannet saab lahendada paljudel eri viisidel, millest mõned on lihtsalt ebaefektiivsed. Samuti tuleb vahel lahendada selliseid ülesandeid mida varem pole vaja läinud. Siis tuleb välja mõelda ka uusi algoritme ja realiseerida neid programmide kujul.

Sorteerimine:

Pane arvud 6, 8, 98, 65, 13, 67, 4, 23, 9, 56, 25, 36, 76, 27, 37, 22, 2, 4, 21, 26, 53 suuruse järjekorda.

Quicksort (sisend-arvud)

Kui sisend-arvude hulgas on 0 või 1 element – on arvud juba sorteeritud; **valmis!**

Vali üks arv sisend-arvudest (nimetame seda arvuks X)

Jaga ülejäänud arvud kahte hulka: väiksemad kui X ja suuremad kui X.

Paiguta X nende kahe hulga vahele nii et väiksemad on vasakul, suuremad paremal

Quicksort (väiksemad)

Quicksort (suuremad)

Näide (rasvaselt on igal sammul valitud X ; see jääb kohe „paigale“)

6, 8, 13, 23, 9, 25, 36, 27, 37, 22, 2, 4, 21, 26, **53**, 98, 65, 67, 56, 76

6, 8, 13, 23, 9, 25, 22, 2, 4, 21, **26**, 36, 27, 37, **53**, 98, 65, 67, 56, 76

6, 8, 13, 9, 2, 4, **21**, 23, 25, 22, **26**, 36, 27, 37, **53**, 98, 65, 67, 56, 76

2, **4**, 6, 8, 13, 9, **21**, 23, 25, 22, **26**, 36, 27, 37, **53**, 98, 65, 67, 56, 76

2, **4**, 6, 8, **9**, 13, **21**, 23, 25, 22, **26**, 36, 27, 37, **53**, 98, 65, 67, 56, 76

2, **4**, 6, 8, **9**, 13, **21**, **22**, 23, 25, **26**, 36, 27, 37, **53**, 98, 65, 67, 56, 76

2, **4**, 6, 8, **9**, 13, **21**, **22**, 23, 25, **26**, 36, 27, **37**, **53**, 98, 65, 67, 56, 76

2, **4**, 6, 8, **9**, 13, **21**, **22**, 23, 25, **26**, **27**, 36, **37**, **53**, 98, 65, 67, 56, 76

2, **4**, 6, 8, **9**, 13, **21**, **22**, 23, 25, **26**, **27**, 36, **37**, **53**, 65, 67, 56, **76**, 98

2, **4**, 6, 8, **9**, 13, **21**, **22**, 23, 25, **26**, **27**, 36, **37**, **53**, **56**, 65, 67, **76**, 98

(Näites ei ole algoritmi lõpuni järgitud kui 2 elementi on juba õiges järjekorras.)

Sellist algoritmi, mis ise-ennast uuesti „välja kutsub“ nimetatakse *rekursiivseks* ja vastavat välja kutsumist *rekursiooniks*. Ülesande jagamist osadeks aga kutsutakse „**Jaga ja valitse**“.

Labürint

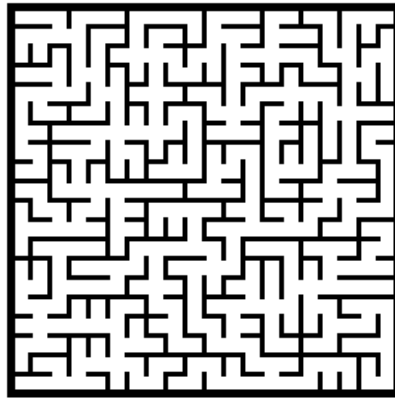
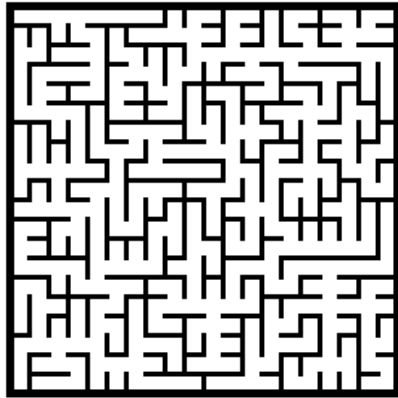
Moodusta $N \times N$ „tuba“ ehk „käiku“

Eemalda juhuslikult seinu tubade X ja Y vahel nii, et tekiks täis labürint:

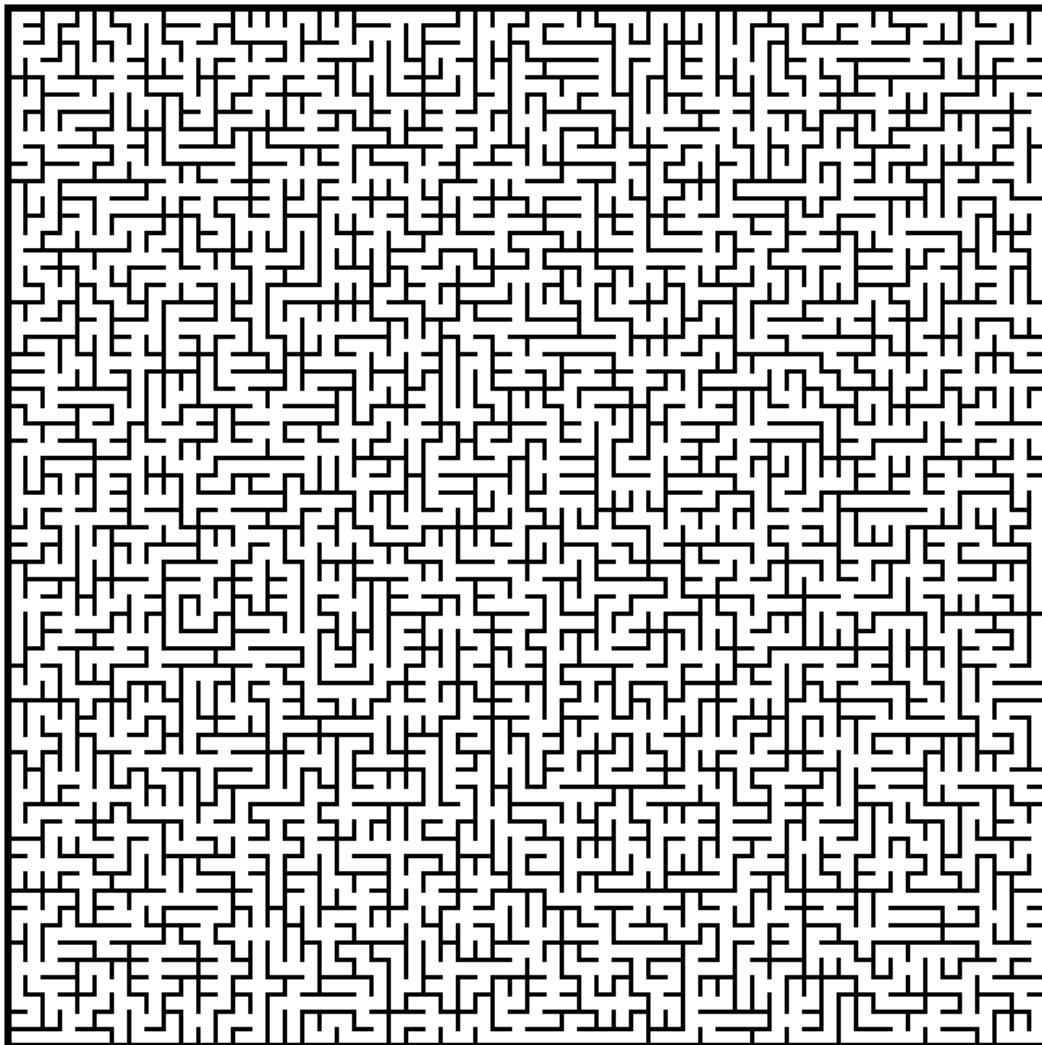
Millistesse käikudesse (tubade hulkadesse) kuuluvad X ja Y?

Seina tubade X ja Y vahel saab eemaldada vaid siis kui X ja Y olid senini erinevates „käikudes“.

Kui eemaldad seinu, **ühenda** „käigud“, ehk kaks senist hulka



20 x 20



60x60

Kõikidest „tubadest“ saab kõikidesse teistesse tubadesse. Otsi üles teed diagonaalide vahel.

<http://emu.at.mt.ut.ee/u/vilo/Algorithmics/maze.cgi>