

1. Multi-Ant (s1)

1 sekund

8 punkti

Suure, paljudest üksteisest sõltuvatest moodulitest koosneva tarkvaratoote kompileerimine ja pakendamine võib olla päris mahukas töö. Ant on programm selle automatiseerimiseks.

Ant saab ette kõigi sammude omavahelised sõltuvused ja teeb vajalikud tegevused õiges järjekorras, see tähendab nii, et ühtegi sammu ei alustata enne kui kõik need, mille tulemustest ta sõltub, on juba ära tehtud.

Paraku on suuremate projektide puhul ka sellest vähe, sest kõigi sammude ühes arvutis järjest tegemine võtab lihtsalt liiga kaua aega. Multi-Ant on Anti edasiarendus eesmärgiga jagada vajalikke tegevusi mitme arvuti vahel.

Kirjutada Multi-Anti komponent, mis saab ette info vajalike sammude omavaheliste sõltuvuste kohta ja koostab plaani nende jagamiseks mitme arvuti vahel nii, et lõpptulemus valmib minimaalse võimaliku ajaga. Iga sammu tegemiseks kulub üks ajaühik ja kui kaks sammu teineteisest ei sõltu, võib neid teha paralleelselt eraldi arvutitel.

Sisend. Tekstifaili `s1sis.txt` esimesel real on vajalike sammude arv N ($1 \leq N \leq 1000$) ja järgmisel N real igaühel info ühe sammu eelduste kohta. See info on antud kujul *samm eeldus1 eeldus2* ..., kus *samm* on sammu enda nimi ja *eeldus1*, *eeldus2*, ... nende sammude nimed, mis peavad olema tehtud enne kui *samm* alata võib. Iga nimi koosneb 1...20 väikesest ladina tähest, kahe kõrvuti oleva nime vahel on alati täpselt üks tühik ja ühegi rea kogupikkus ei ületa 250 märki.

Väljund. Tekstifaili `s1val.txt` esimesele reale väljastada kõigi sammude tegemiseks vajalik aeg T ja järgmisele T reale tegevusplaani ajaühikute kaupa. Faili teisele reale väljastada tühikutega eraldatult esimesel ajaühikul paralleelselt tehtavate sammude nimed, kolmandale reale teisel ajaühikul tehtavad sammud j.n.e. Võib eeldada, et avuteid on piiramatult koguses — vajadusel ostab firma neid juurde. Kui minimaalse ajaga plaane on mitu väljastada ükskõik milline neist.

Näide.

<code>s1sis.txt</code>	<code>s1val.txt</code>
8	6
checkout clean	init
clean init	clean
compile checkout clean	checkout
coverage compile	compile
deploy test	test coverage
init	deploy report
report coverage	
test compile	