

**Lygiagretieji algoritmai**  
3 laboratorinis darbas  
Susijusių sričių paveikslėlyje identifikavimas  
Vyktas Kumštys

**Duota:** binarinė paveikslė išraiška (0 - background, 1 - image)

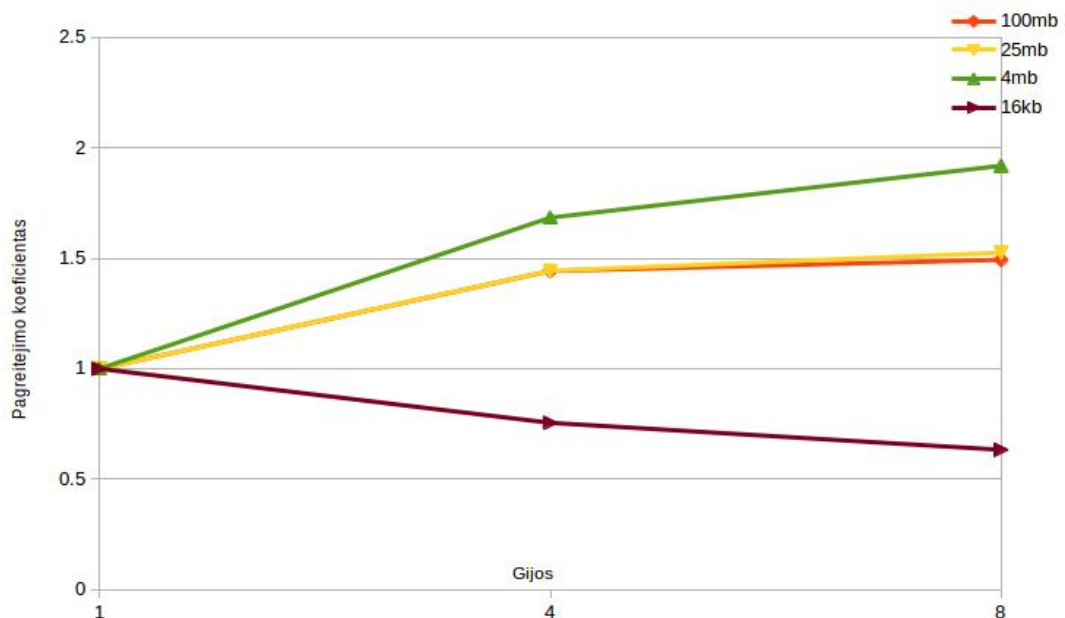
**Užduotis:** Pažymėti kiekvieną teritoriją (paveikslė atskiri objektai turės skirtingus skaičius)

**Rezultatas:** Paveikslė objektai bus atskirti

**Sprendimas:**

Paveikslą turime dvimačiame masyve. Galimi lygiagretumo taikymai naudojant 4 arba 8 gijas. Prieš pradedant algoritmo taikymą suskirstome masyvą į tiek apylygių dalių, kiek gijų naudosisime (1, 4 arba 8). Kiekviena gija turės savo teritoriją, kurią pereis taikant algoritmą. Kadangi algoritmas veikia rekursiškai (nagrinėjami taško kaimynai), gali atsitikti taip, kad bus nueinama iš savo paskirto ploto ribų. Tokiu atveju pikselis bus žymimas didesniu skaičiumi (tokie susikirtimai atima vykdymo laiką, bet bendrose situacijose tai nėra esminis faktorius).

Testavimas buvo atliktas su 4 skirtingų dydžių paveikslais. Kiekvienas bandymas buvo atliktas 100 kartų ir gaunamas vidutinis 1 bandymo laikas. Taip ant kiekvieno paveikslė buvo bandoma 3 skirtingais režimais (1, 4 bei 8 gijos). Gauti rezultatai rodomi grafiku:



Iš grafiko matome, kad su didesniais failais išlošiamo laiko nuo gijų skaičiaus (kuo gijų daugiau, tuo greičiau įvykdysime algoritmą). Įdomu pastebėti, kad geriausi rezultatai gauti su 4mb dydžio failu (pagreitinimas beveik 2 kartus). Taip gali būti dėl to, kad faile buvo labai dėkingas vienetukų (image) išsidėstymas. Nebuvo daug susidūrimų, kurių dėka

reikėtų pereit po antrą kartą per tą pačią vietą. Taip pat gali būti, kad background procentaliai buvo daugiau nei pačio image lyginant su kitais failais. Faktas, kad mažiems failams taikyti gijų visiškai neapsimoka. Mažus failus puikiai pažymėti galime ir netaikant lygiagretumo. Galime daryti prielaidą, kad dydinant gijų skaičių, failo dydį, galime išlošti ir dar daugiau laiko.