Atsiskaitymai už semestro darbus (pratybos)

Trečias kontrolinio perrašymas egzamino metu (nepamirškite gauti leidimą dekanate).

Egzamino tvarka (2016):

Egzamino užduotį sudarys du teoriniai klausimai (po 3 balus) ir dvi praktinės užduotys (po 2 balus).

Teoriniai klausimai pateikti žemiau. Praktinės užduotys iš viso kurso. Prie klausimų pateikta nuorodos į vadovėlio THOMAS H. CORMEN, et al. Introduction to algorithms. Cambridge: MIT press. 2 leidimą.

Egzaminą pradės 8-10 studentų, kurie gavę užduotį ruošis apie 45min. po to laikys egzaminą žodžiu tokia pačia eilės tvarka kaip traukė bilietą. Egzaminą išlaikiusius keis nauji, kurie vėl galės ruoštis apie 45 min.

Egzamino klausimai:

I egzamino klausimo variantai

1. Funkcijų augimo asimptotiniai žymėjimai ir jų apibrėžimai. (3.1 sk.; 41–50 psl.)
2. Rekurentinių sąryšių sprendimo būdai (aprašyti idėją). Suformuluoti Pagrindinę teoremą. 4.1, 4.2, 4.3 sk.; 62–75 psl.)
3. Dekompozicinių algoritmų sudėtingumo T(n) skaičiavimo formulės struktūra ir sprendimo būdai.
4. Rikiavimas su įterpimu (Insert sort). Įrodyti algoritmo korektiškumą ir jo sudėtingumą kai duomenys įvedami nepalankiausiu atveju. (2.2 sk.; 23–25 psl.)
5. Rikiavimas su įterpimu (Insert sort). Įrodyti algoritmo korektiškumą ir jo sudėtingumą kai duomenys įvedami palankiausiu atveju (2.2 sk.; 23–25 psl.).
6. Rikiavimo algoritmo suliejimo (Merge sort) būdu korektiškumo įrodymas.(2.3.1 sk.; 28–32 psl.)
7. Rikiavimo algoritmo suliejimo (Merge sort) būdu sudėtingumo įrodymas. (2.3.2 sk.; 32–36 psl.)
8. Duomenų struktūra – „piramidė“? Kaip priklauso piramidės dydis ir aukštis nuo rikiuojamų duomenų kiekio? Piramidės savybių palaikymas: procedūra Max-Heapify ir jos sudėtingumo įvertinimo įrodymas. (6.1, 6.2 sk.; 126–134psl.)
9. Kaip atliekamas piramidės sutvarkymas (pateikite algoritmą ir sudėtingumą įrodykite)? Rikiavimo piramide (Heap sort) algoritmo idėja ir sudėtingumo viršutinis įvertinimo įrodymas. (6 sk.; 6.3, 6.4 sk.; 126–134psl.)
10. Greito rūšiavimo (Quick sort) algoritmas ir jo korektiškumo analizė. Įrodykite algoritmo sudėtingumą blogiausiu atveju. (7.1 sk. 145–148 psl.)
11. Greito rikiavimo (Quick sort) algoritmo idėja. Kada greito rikiavimo algoritmo sudėtingumas ir rikiavimo piramide (Heap sort) sudėtingumo asimptotiniai įverčiai konstantos tikslumu sutampa? Įrodykite kokiam nors atvejui (rūšiavimo piramide sudėtingumo asimptotinio įverčio įrodyti nereikia). (7.2 sk. 150–151 psl.)
12. Įrodykite greito rikiavimo (Quick sort) sudėtingumą geriausiam ir blogiausiam atvejui. (7.2 sk. 150 psl.; 7.4.1 sk. 155 psl.)
13. Optimalūs rūšiavimo algoritmai naudojantys palyginimus. Įrodyti, kad rūšiavimas piramide (Heap sort) ir suliejimo (Merge sort) būdu yra asimptotiškai optimalūs algoritmai. (8.1 sk.; 165–167psl.)
14. Tiesiniai rūšiavimo algoritmai: rikiavimas skaičiuojant (Counting sort) ir pozicinis (Radix sort). Kodėl jie lenkia asimptotiškai optimalius rikiavimo algoritmus? (8.2, 8.3 sk.; 168–173 psl. teiginiai iš 8.3 sk. neprivalomi)
15. Kišeninis rikiavimas (Bucket sort). Jo sudėtingumo įvertinimo įrodymas. (8.4sk; 174–175 psl. iki 8.1 formulės)

16. Tiesioginio adresavimo lentelės. Maišos lentelės (suformuluoti uždavinius ir pateikti veiksmų algoritmus ir jų sudėtingumo įvertinimus pagrįsti). Koki šio būdo trūkumai. (11.1 sk. 222 psl.; 11.2 sk. 224psl.)
17. Paprastas tolygus „hešavimas“. Suformuluoti teoremas apie paieškos laiko įvertinimą. Įrodyti vieną teoremą. (11.2 sk. 225–228psl., mokėti įrodyti vieną iš teiginių 11.1 ir 11.2)
18. Atviras adresavimas. Suformuluoti uždavinį, pateikti veiksmų algoritmus ir jų sudėtingumo įvertinimus pagrįsti. Kada tikslinga naudoti? Atviro adresavimo maišos funkcijų pavyzdžiai. (11.4 sk. 237–241 psl.)

II egzamino klausimo variantai

1. Paieškos binarinis medis. Struktūra. Operacijos ir jų sudėtingumo įrodymai.(12.1 sk. 253–255 psl. iki 12.1 teiginio įrodymo; 12.2 sk. 256–259 psl.)
2. Juodai raudoni paieškos binariniai medžiai. Įrodykite teoremą apie paieškos medžio aukštį. (13.1 sk. 273–276 psl.)
3. Dinaminio programavimo taikymas sprendžiant Surinkimo linijos planavimo uždavinį. (15.1 sk. 324–330 psl.)
4. Dinaminio programavimo taikymas optimizuojant matricų daugybą. (15.2 sk. 331–338 psl.)
5. Dinaminio programavimo elementai. (15.3 psl. 339–347psl. 345 psl. esančio algoritmo ir jo sudėtingumo radimo 346 psl. nereikia)
6. Godūs algoritmai. Maksimalios procesų aibės radimo (Procesų pasirinkimo) uždavinys ir jo sprendimas taikant godžią procesų pasirinkimo strategiją. (16.1 sk. 371–375psl.)
7. Amortizacinė algoritmų analizė. (17sk. žinoti metodų idėjas ir pateikti pavyzdžius)
8. Duomenų struktūros aprašančios grafą ir jų privalumai bei trūkumai. Paieškos į plotį algoritmas ir jo sudėtingumo įvertinimas. (22.1 sk. 527–529 psl. 22.2 sk. 531–534 psl.)
9. Paieškos į gylį algoritmas ir jo sudėtingumo įvertinimas. Grafo briaunų klasifikavimas. Teiginys apie paieškos į gylį grafo briaunas. (22.3 sk. 540–543 psl. iki paieškos į gylį savybių; 546–547psl.)
10. Topologinis rikiavimas bei stipriai susietų komponentų paieškos algoritmai bei jų sudėtingumas. (22.4 sk. 549–550 psl. iki teiginių; 22.5 sk. 552–554 psl. iki teiginių)
11. Minimalūs dengiantys medžiai. Jų radimo algoritmų sudarymo metodika (Bendras algoritmas). Teiginio apie lengviausią briauną įrodymas. (23.1 sk. 562–565 psl. 23.1 teiginys)
12. Kruskalo algoritmas. Sudėtingumo įrodymas. Algoritme naudojamų aibių struktūra ir veiksmų su jomis sudėtingumo įvertimas (23.2 sk. 567–570 psl.).
13. Prima algoritmas. Sudėtingumo įrodymas. Algoritme naudojamų struktūros ir veiksmų su jomis sudėtingumo įvertimas. (23.2 sk. 570–573 psl.)
14. Uždavinio trumpiausių kelių iš vienos viršūnės radimo sprendimas. Belmano–Fordo algoritmas ir jo sudėtingumo įvertinimas. (24 sk. 580–586 psl.; 24.1 sk. 588 psl.)
15. Trumpiausių kelių iš vienos viršūnės paieška acikliniame grafe bei Deikstra algoritmas ir jų sudėtingumų įvertinimai. (24.2 sk. 592; 24.3 sk. 595–599 psl. Teiginių apie algoritmų korektiškumus ir jų įrodymų nereikia)
16. Trumpiausių kelių paieška tarp visų viršūnių taikant dinaminį programavimą. Algoritmo sąsaja su matricų daugyba. Algoritmo sudėtingumo įvertinimas. (25.1 sk. 622–627 psl.)
17. Fordo-Fulkersono metodas. Liekamasis tinklas. Srautą didinantis kelias. Fordo-Fulkersono ir Edmontono-Karpo algoritmai ir jų sudėtingumai (26.2 sk. 651 psl., 564 psl., 657–660 psl.)
18. NP sudėtingumas. P ir NP klasės. Uždavinių pavyzdžiai. (34 sk. 966–969 psl.)

Dalius Makackas