

Keerukamad graafialgoritmid

Vähim toeseppuu (Minimal Spanning Tree)

Antud kaalutud servadega suunamata sidus graaf. Leida minimaalse servapikkuste summaga puu, mis sisaldab kõiki graafi tippe.

Primi Algoritm

Vali suvaline tipp toeseppuu alguseks.

Kuni leidub veel tippe, mis puusse pole lisatud,
leia lühim serv, mis seob mõnda puu tippu seal mitteolevaga,
lisa see serv puusse.

Realisatsioon: Pidada iga tipu kohta meeles odavaimat viisi tema lisamiseks puusse. Tipu lisamisel puusse vaatata üle ta naabrid ja vajadusel nendel seda väärtust korrigeerida. Kahendpuud kasutades on väga sarnane Dijikstra algoritmile.

Keerukus: Kui minimaalse kauguse leidmine on tehtav $O(1)$ ajaga, on keerukus $O(n \log n)$.

Eelised: Väga sarnane Dijikstra algoritmile.

Kruskali algoritm

Alusta tühjast puust.

Kuni lisatud on vähem kui $n-1$ serva,
lisa lühim serv nende hulgast, mis ei tekita eelnevatega koos tsükli.

Realisatsioon: Sorteerida alguses servad kasvavas järjekorras. Iga tipu juures pidada meeles tema "ülemat", mis alguses ta ise. Kahe tipu sidumisel kontrollida, kas nende "ülemülemad" (st ülemate ahelat mööda lõppu välja jõudes saadavad isendid) kattuvad. Kui ei, siis sea neist ühe ülemülem teise omaks. Kui jah, liigu edasi järgmise serva juurde, kuna see moodustaks tsükli, sest tema otstipud on juba mingil viisil seotud.

Keerukus: $O(n \log n)$, tegelikult praktiliselt $O(n)$, kui kasutada veel teatud trikke. (Välja arvatud servade sorteerimine, mis on $O(m \log m)$.)

Eelised: Suhteliselt lihtne kirjutada, kuna ei vaja väga keerulist andmestruktuuri. Toimib ka mittesidusa graafi puhul, kui modifitseerida veidi korduse tingimust.

Euleri ahela/tsükli leidmine (Eulerian path/tour)

Antud sidus (suunatud või suunamata) graaf. Leida graafis ahel/tsükkel, mis läbib kõiki servi täpselt ühe korra.

Algoritm

Kontrollida, kas graafis on üldse nõutud ahel või tsükkel:

Vaadata, kas kõikidesse tippudesse läheb sisse sama palju servi kui sealt välja (suunamata servade korral, et iga tipu juures oleks paarisarv servi). Kui vaja vaid ahelat, võib ühte tippu tulla sisse ja ühest minna välja üks serv vähem.

Ahela leidmine:

Alustada ühest paaritu arvulise servade arvuga tipust (Tsükli puhul suvalisest tipust).

Kuni tipust saab kuhugi edasi liikuda, kustuta edasi viiv serv ja mine uude tippu.

Kui ei saa, lisa tipp otsitava ahela/tsükli lõppu ja mine tulnud teed üks samm tagasi.

Realisatsioon: Rekursioon on siinkohal mugav (uues tipus kutsuda välja sama funktsiooni). Suuremate graafide korral võib nii paraku mälust puudu tulla, misjuhul tuleb rekursiooni ise emuleerida kahe magasinini (stack) abil.

Keerukus: $O(m)$

Maksimaalne voog (Maximum flow)

Antud kaalutud servadega (suunatud või suunamata) graaf, kus serva kaal kujutab tema "läbilaskevõimet", alg Tipp ja lõpptipp. Graaf kujutab torustikku. Soovitakse teada maksimaalset võimalikku seda korruga läbiva voolu hulka.

Ford-Fulkersoni algoritm

Leia servade arvu poolest lühim tee, mida mööda saab veel voogu alg tipust lõpptippu viia.

Leia vähima läbilaskevõimega serv sellel teel.

Vähenda tee iga serva läbilaskevõimet leitud serva läbilaskevõime võrra ning suurendada sama palju nende servadega vastassuunas liikuvate servade läbilaskevõimeid.

Korrata eelnevat, kuni selline tee leidub.

Realisatsioon: Iga serva hoida mõlemas suunas (parim maatriksesitus).

Keerukus: $O(n \cdot F)$, kus F on leitud maksimaalne voog, või $O(n \cdot m^2)$, kus m on servade arv (tegelikult enamasti palju parem).