

GRÁFOK ÉS ALGORITMUSOK ELMÉLETE

Matematika BSc Elemző szakirány II. év 1. félév

Az előadáson a következő fejezetek kerültek ismertetésre:

1. Algoritmusokról (leírási mód, futási idők, aszimptotika: az Ordó matematikája)
Verembe, Veremből műveletek (tömbös) – $O(1)$
Maximum kiválasztás – $O(n)$
Buborékrendeztetés – $O(n^2)$
Barkochba – $O(\log n)$ (Pszudokód helyett döntési fa szerepel.)
Hanoi tornyai – $O(2^n)$
További nagyságrendekre majd jönnek algoritmusok; pl. az $O(n \log n)$ -es rendezések.
2. Verem
Műveletek (tömbös ábrázolás): ÜresVerem, Üres?, Tele?, Verembe, Veremből, Felső.
Alkalmazás: helyes zárójelzés feldolgozása (összetartozó nyitó-cukó párok felismerése veremmel).
3. Sor
Műveletek (tömbös ábrázolás): ÜresSor, Üres?, Tele?, Sorba, Sorból, Első.
Alkalmazás: bináris fa szintfolytonos bejárása (lásd a bináris fánál).
4. Listák
A láncolt lista, mint ábrázolási mód, pl. verem, sor pointeres megvalósítása, műveleteikkel együtt (itt a Tele? nem szerepel).
Alkalmazás: beillesztő rendezés fejelemes listára (lásd a rendezéseknél).
Lista, mint adattípus: fejelem van/nincs, egyirányú/kétirányú, ciklikus/nem-ciklikus.
Pointer-szintű algoritmus: egyszerű lista megfordítása helyben.
A fejelemes egyszerű listának a teljes műveleti csomagja: ÜresLista, Üres?, Elsőre, Következőre, Utolsó?, Vége?, AktÉrték, AktMod, BeszúrElsőnek, BeszúrUtán, TörölAkt.
Alkalmazása: lista szekvenciális feldolgozása.
5. Bináris fa
Láncolt ábrázolás. Tömbös ábrázolás (teljes, illetve majdnem teljes, balra rendezett bináris fa), szülő – gyerek csúcsok indexfüggvényei.
Három bejárás: preorder, inorder, posztorder.
Bináris fa szintfolytonos bejárása (sor adatszerkezet használatára példa).
Rekurzív függvények bináris fákban, pl. levelek számának meghatározása.
6. Kupac, elsőbbségi sor
Kupac, mint majdnem teljes balra tömörített bináris fa (ennek magassága), a szülő-gyerekei csúcsokban található értékek összefüggése.
Kupac ábrázolása tömbben (emlékeztető: szülő-gyerek indexfüggvények).
Kupac építése egy tömbben elhelyezett input értékekből.
Kupac műveletei: maximum (vagy minimum) törlése, új kulcs beszúrása (csak végrehajtás adott ábrán).
Elsőbbségi sor (megvalósítható rendezetlen vagy rendezett tömbbel is, de általában a kupacot választjuk), műveletei: ÜresPrSor, Üres?, PrSorba, PrSorból, MaxPrior.
7. Keresések és kiválasztások
Maximum kiválasztás (tömbös)
Barkochba

Bináris keresés (két változat: $<$, $=$, $>$ ágakkal, illetve $\leq$, $>$ ágakkal (ez végigkeres és rákérdez);

Két legnagyobb elem kiválasztása (GY)

Legnagyobb és legkisebb elem kiválasztása egyszerre (GY)

A k -adik elem kiválasztása véletlenített algoritmussal (várhatóan lineáris időben) (GY)

8. Bináris keresőfa

Fogalma, min. és max. magassága, átlagos magassága ($\leq 1,39 \cdot \log n$, levezetés nélkül).

Nevezetes tulajdonság: inorder bejárással rendezett kulcssorozatot kapunk.

Műveletek: Keres, MinKulcs, Következő (csak végrehajtás adott ábrán), Beszúr, Töröl (csak végrehajtás adott ábrán).

9. Az AVL-fa

Fogalma, magassága ($\leq 1,44 \log n$, levezetéssel).

Forgatások beszúrás esetén (csak végrehajtásuk adott ábrán).

A forgatások tulajdonságai: helyesek; visszaáll a beszúrás előtti famagasság; beszúrásnál elég egy is.

Az AVL-fa építésének műveletigénye (beszúrás, ellenőrzés, forgatás)

10. Összehasonlító rendezések (műveletigény mindegyiknél, levezetés nélkül)

Buborék rendezés

Maximumkiválasztó rendezés

Beillesztő rendezés (tömbre és láncolt listára is)

Versenyrendezés elve

Kupacrendezés (csak végrehajtás ADS szinten)

Gyorsrendezés (a Particionál eljárás „az ötösért”)

Összefésülő rendezés (az Összefésülő eljárás: nem)

Az összehasonlító rendezések időbeli alsókorlátja (egyelőre csak kimondtuk)

11. Gráfok ábrázolásai

Gráfelméleti alapfogalmak ism.: $G=(V, E)$ gráf, irányított és irányítatlan gráf, $c: E \rightarrow R$ élsúly, egyszerű gráf: nincs hurokél és többszörös él; szomszéd, leszármazott, út, kör, körmentes gráf, fokszám, összefüggő, ill. erősen összefüggő gráf.

Gráf ADS = matematikai gráf-fogalom. Gráf adatszerkezet ábrázolása szomszédsági mátrixszal, illetve éllistával; élsúlyozott gráf esete, helyfoglalás irányított és irányítatlan gráf esetén, ajánlott alkalmazásuk: az élszám nagyságrendjétől függően.

12. Szélességi bejárás

A szélességi bejárás/keresés stratégiája. A feladat pontos megfogalmazása.

A szélességi bejárás algoritmusa: költség és szülő tömb, sor adatszerkezet, három szín (= nem-látott, látott, bejárt csúcs).

Műveletigény elemzése.

13. Minimális költségű utak egy forrásból I. (Dijkstra-algoritmus)

Adott kezdőcsúcsból a gráf minden csúcsába minimális költségű út keresése nemnegatív élköltségek esetén; a feladat pontos megfogalmazása. A megoldás mohó módszere.

A Dijkstra-algoritmus első megfogalmazása: iterált feltételes minimum-választás a költség-tömb „nem-kész” elemei közül.

A Dijkstra-algoritmus második, absztraktabb megfogalmazása: minimum választó elsőbbségi sor bevezetése, „kész” halmaz használata.

Hogyan származtatható ebből az előző, illetve a kupacos változat?

Műveletigény elemzése mindkét esetben, illetve külön a „sűrű” gráf esetére ($e = \Theta(n^2)$)

14. Minimális költségű feszítőfák (Prim-algoritmus)
Szemléletes alkalmazás: vezetékhálózat behúzása több helységet tartalmazó tájon (pl. villanyáram, gáz). A feladat pontos megfogalmazása.
A megoldások közül mohó jellegű Prim-algoritmust választjuk és a Dijkstra-algoritmus absztrakt változatából származtatjuk a költségekre vonatkozó kérdés módosításával.
Műveletigény.
15. Minimális költségű utak egy forrásból II. (Bellman-Ford-algoritmus)
Negatív költség megengedett, de a negatív kör: nem. Példa arra, hogy nem jó a mohó algoritmus.
Bellman-Ford-algoritmus: $n-1$ iterációban minden élen közelítést végzünk. Egy n -edik ellenőrző iteráció a negatív körök kizárására.
Az élek sorrendje az iteráción belül kihat a ténylegesen szükséges iterációkra.
Műveletigény.
16. Mélységi bejárás, élek osztályozása
A mélységi bejárás stratégiája: előrehaladás, ameddig lehet, majd visszalépés.
Jellemzője, hogy nincs kitüntetett kezdőcsúcs és minden csúcsot bejár.
A feladat pontos megfogalmazása. Belépési és kilépési számok a csúcsoknál.
Éltípusok. Az algoritmus rekurzív és színeket használ.
Az éltípus (pl. visszaél) meghatározásának helye. Műveletigény.
17. DAG topologikus rendezése
A DAG: irányított körmentes gráf. Állítás: G DAG akkor és csak akkor, ha tetszőleges MB talál visszaélet.
Def.: G gráf topologikus rendezése. Lemma: Ha G DAG, akkor van olyan csúcsa, amelybe nem fut be él.
Állítás: G DAG akkor és csak akkor, ha G -nek van topologikus rendezése. Az akkor-irány konstruktív bizonyítása alapján készíthető algoritmus.
A topologikus rendezés algoritmusa MB alkalmazásával, a kilépési számok verembe írásával. Műveletigény.
18. Hashelés (csak áttekintés)
Kulcsütközés feloldása láncolással, illetve nyílt címzéssel
Hash-függvények
19. Rendezés lineáris időben (áttekintés, csak végrehajtás)
Leszámoló rendezés
Edényrendezés, kiterjesztés („100 dolgozat” példa)
Számjegypozíciós rendezés „előre”
Számjegypozíciós rendezés „visszafelé”
Kulcsmezők szerinti „visszafelé haladó” rendezés listákkal
20. Dinamikus programozás (gyakorlaton!)
Fibonacci számok
Hátizsák feladat
Mátrixszorzás zárójelezése
21. Számolás nagy számokkal / Moduláris aritmetika / Az RSA alapjai (önálló tanulásra; nem vizsgaanyag)
Euklideszi algoritmus; kiterjesztett euklideszi algoritmus;
mod m hatványozás
RSA alapjai