Algoritmusok és adatszerkezetek II. előadásjegyzet:

Bevezetés, Jelölések, Tematika

Ásványi Tibor – asvanyi@inf.elte.hu

2021. november 19.

Tartalomjegyzék

1. Bevezetés, Ajánlott irodalom

Kedves Hallgatók!

A vizsgára való készülésben elsősorban az előadásokon és a gyakorlatokon készített jegyzeteikre támaszkodhatnak. További ajánlott források:

Hivatkozások

[1] Ásványi Tibor, Algoritmusok és adatszerkezetek II.
Útmutatások a tanuláshoz, Tematika, fák, gráfok, mintaillesztés, tömörítés
http://aszt.inf.elte.hu/~asvanyi/ad/ad2jegyzet/

[2] Ásványi Tibor, Algoritmusok II. gyakorló feladatok (2016)
http://aszt.inf.elte.hu/~asvanyi/ad/ad2jegyzet/ad2feladatok.pdf

[3] Cormen, T.H., Leiserson, C.E., Rivest, R.L., Stein, C.,
magyarul: Új Algoritmusok, *Scolar Kiadó*, Budapest, 2003.
ISBN 963 9193 90 9
angolul: Introduction to Algorithms (Third Edititon),
The MIT Press, 2009.

[4] Fekete István, Algoritmusok jegyzet
http://ifekete.web.elte.hu/

[5] Rónyai Lajos – Ivanyos Gábor – Szabó Réka, Algoritmusok,
TypoTEX Kiadó, 1999. ISBN 963 9132 16 0
https://www.tankonyvtar.hu/hu/tartalom/tamop425/2011-0001-526_ronyai_algoritmusok/adatok.html

[6] Tarjan, Robert Endre, Data Structures and Network Algorithms,
CBMS-NSF Regional Conference Series in Applied Mathematics, 1987.

[7] Weiss, Mark Allen, Data Structures and Algorithm Analysis,
Addison-Wesley, 1995, 1997, 2007, 2012, 2013.

[8] Carl Burch, Ásványi Tibor, B+ fák
http://aszt.inf.elte.hu/~asvanyi/ad/ad2jegyzet/B+ fa.pdf

[9] Ásványi Tibor, Algoritmusok és adatszerkezetek I. előadásjegyzet (2021)
http://aszt.inf.elte.hu/~asvanyi/ad/ad1jegyzet/ad1jegyzet.pdf

[10] Wirth, N., Algorithms and Data Structures, *Prentice-Hall Inc.*, 1976, 1985, 2004.
magyarul: Algoritmusok + Adatstruktúrák = Programok, *Műszaki Könyvkiadó*, Budapest, 1982. ISBN 963 10 3858 0

Ezúton szeretnék köszönetet mondani *Umann Kristófnak* az AVL fákról szóló fejezetben található szép, színvonalas szemléltető ábrák elkészítéséért, az ezekre szánt időért és szellemi ráfordításért!

A vizsgákon az elméleti kérdések egy-egy tétel bizonyos részleteire vonatkoznak. Lesznek még megoldandó feladatok, amik részben a tanult algoritmusok működésének szemléltetését, bemutatását, részben a szerzett ismeretek kreatív felhasználását kérik számon. Egy algoritmus, program, művelet bemutatásának mindig része a műveletigény elemzése.

Az előadások elsősorban a CLRS könyv [3] (ld. alább) angol eredetijének harmadik kiadását követik, de pl. a piros-fekete fák helyett az AVL fákat tárgyaljuk, a jelöléseket az első féléves jegyzetnek megfelelően módosítva. (A CLRS könyv [3] érintett fejezeteiben a magyar és az angol változat között leginkább csak néhány jelölésbeli különbséget találtunk.)

Az egyes struktogramokat általában nem dolgozzuk ki az értékadó utasítások szintjéig. Az olyan implementációs részleteket, mint a listák és egyéb adatszerkezetek, adattípusok műveleteinek pontos kódja, a dinamikusan allokált objektumok deallokálása stb. általában az Olvasóra hagyjuk, hiszen ezekkel az előző félévben foglalkoztunk. Használni fogunk olyan absztrakt fogalmakat, mint a véges halmazok, sorozatok, gráfok. Ezeket, ha valamely eljárás paraméterlistáján szerepelnek – mint strukturált adatokat – minden esetben cím szerint vesszük át. (A skalárok változatlanul érték vagy cím szerint adódnak át, ahol az érték szerinti paraméterátvétel az alapértelmezett.)

2. Jelölések

A struktogramokban a „for" ciklusok (illetve a „for each" ciklusok) mintájára alkalmazni fogunk a ciklusfeltételek helyén pl. „$\forall x : P(x)$" alakú kifejezéseket, ami azt jelenti, hogy a ciklusmagot a $P(x)$ állítás igazsáhalmazának minden x elemére végre kell hajtani, valamint „$\forall v \in V$" alakúakat, ami a „$\forall v : v \in V$" rövidítése. Ehhez $P(x)$ igazsághalmazának, illetve V-nek végesnek, és hatékonyan felsorolhatónak kell lennie. Ez ideális esetben azt jelenti, hogy a felsorolás műveletigénye az igazsághalmaz, illetve V méretétől lineárisan függ.

Ha a ciklus fejében „$i := u$ to v" alakú kifejezést látunk, akkor a ciklusmag az $u..v$ egész intervallum i elemeire szigorúan monoton növekvő sorrendben hajtódik végre.

Ha pedig a ciklus fejében „$i := u$ downto v" alakú kifejezést látunk, akkor a ciklusmag a $v..u$ egész intervallum i elemeire hajtódik végre, de szigorúan monoton csökkenő sorrendben.

A fentiek szerint az egyszerűbb programrészletek helyén gyakran szerepelnek majd magyar nyelvű utasítások, amiknek részletes átgondolását, esetleges kidolgozását, a korábban tanultak alapján, szintén az Olvasóra bízzuk. A struktogramokban az ilyen, formailag általában felszólító mondatok végéről a felkiáltójelet elhagyjuk (mivel az adott szövegkörnyezetben ez gyakran faktoriális függvényként lenne [félre]érthető).

3. Tematika

Minden tételhez: Hivatkozások: például a „[3] 8.2, 8.3, 8.4” jelentése: a [3] sorszámú szakirodalom adott (al)fejezetei.

1. Veszteségmentes adattömörítés. Naiv módszer. Huffman kód, kódfa, optimalitás. LZW (Lempel-Ziv-Welch) tömörítés. [1]

2. Általános fák, bináris láncolt reprezentáció, bejárások ([1]; [10] 4.7 bevezetése). Egyéb reprezentációk? (HF)

3. AVL fák és műveleteik: kiegyensúlyozási sémák, programok. Az AVL fa magassága ([1], [4] 12; [10] 4.5).

4. B+ fák és műveleteik [8].

5. Elemi gráfalgoritmusok ([3] 22). Gráfábrázolások (representations of graphs).

A szélességi gráfkeresés (breadth-first search: BFS). A szélességi gráfkeresés futási ideje (the run-time analysis of BFS). A legrövidebb utak (shortest paths). A szélességi feszítőfa (breadth-first tree). HF: A szélességi gráfkeresés megvalósítása a klasszikus gráfábrázolások esetén; hatékonyság.

A mélységi gráfkeresés (depth-first search: DFS). Mélységi feszítő erdő (depth-first forest). A gráf csúcsainak szín és időpont címkéi (colors and timestamps of vertexes). Az élek osztályozása (classification of edges, its connections with the colors and timestamps of the vertexes). A mélységi gráfkeresés futási ideje (the run-time analysis of DFS). Topologikus rendezés (topological sort). HF: A mélységi gráfkeresés és a topologikus rendezés megvalósítása a klasszikus gráfábrázolások esetén; hatékonyság.

6. Minimális feszítőfák (Minimum Spanning Trees: MSTs). Egy általános algoritmus (A general algorithm). Egy tétel a biztonságos élekről és a minimális feszítőfákról (A theorem on safe edges and MSTs). Prim és Kruskal algoritmusai (The algorithms of Kruskal and Prim). A futási idők elemzése (Their run-time analysis). HF: A Prim algoritmus implementációja a két fő gráfábrázolás és a szükséges prioritásos sor különböző megvalósításai esetén (The implementations of the algorithm of Prim with respect to the main graph representations and representations of the priority queue).

7. Legrövidebb utak egy forrásból (Single-Source Shortest Paths). A legrövidebb utak fája (Shortest-paths tree). Negatív körök (Negative cycles). Közelítés (Relaxation).

A sor-alapú (Queue-based) Bellman-Ford algoritmus. A menet (pass) fogalma. Futási idő elemzése. Helyessége. A legrövidebb út kinyomtatása.

Legrövidebb utak egy forrásból, körmentes irányított gráfokra. (DAG shortest paths.) Futási idő elemzése. Helyessége.

Dikstra algoritmusa. Helyessége. Fontosabb implementációi a két fő gráfábrázolás és a szükséges prioritásos sor különböző megvalósításai esetén. A futási idők elemzése.

8. Legrövidebb utak minden csúcspárra (All-Pairs Shortest Paths). A megoldás ábrázolása a (D, Π) mátrix-párral. HF: Adott csúcspárra a legrövidebb út kinyomtatása.

A Floyd-Warshall algoritmus és a $(D^{(k)}, \Pi^{(k)})$ mátrix párok. A futási idő elemzése. Összehasonlítás a Dijkstra algoritmus, illetve (HF:) a sor-alapú Bellman-Ford algoritmus $|G.V|$-szeri végrehajtásával.

Irányított gráf tranzitív lezártja (Transitive closure of a directed graph) és a $T^{(k)}$ mátrixok. Az algoritmus és futási ideje. HF: összehasonlítás a szélességi keresés $|G.V|$-szeri végrehajtásával.

9. Mintaillesztés (String Matching). Egy egyszerű mintaillesztő algoritmus (The naive string-matching algorithm). A futási idő elemzése.

A Quick Search algoritmus. Inicializálása. A futási idő elemzése.

A Knuth-Morris-Pratt algoritmus. Inicializálása. A futási idő elemzése.