

DMA 2021 uge 11, udvalgte opgaveløsninger

Rasmus Pagh

Løsningerne er skrevet ned meget kortfattet. Her må man selv lave “mellem-regningerne”. Hensigten er at man kan tjekke at man ikke er helt ved siden af. Der kan også have sneget sig fejl ind. Hvis du mener der er en fejl, så rapportér det på Discussions på kursussiden, så det kan blive afklaret og andre kan få glæde af det, hvis du har ret.

Mandag

1. CLRS 17.4-1. Det er nødvendigt at øge kapaciteten når load factoren nærmer sig 1, ellers bliver søgetiden stor. Hvis vi gør det når load factor er lig med $\alpha < 1$, er den forventede søgetid altid $O(1/(1 - \alpha))$. For at implementere dette er hashtabellen nødt til at holde styr på det aktuelle antal elementer n og tabellens størrelse m . Ved indsættelse checkes om $n > \alpha m$, og i givet fald øges fra m til m' (fx fordobling, $m' = 2m$). Det sker ved at alle elementer indsættes i en ny hashtabel af størrelse m' . Hvis $m = \Theta(n)$ er omkostningen ved dette $O(n + m)$, som kan amortiseres ud over de $\Omega(n)$ operationer, der er mellem to udvidelser. Da udvidelserne sker på bestemte tidspunkter, bestemt af antal nøgler, vil visse operationer ikke længere have forventet konstant køretid.
2. **Multi-delete.** Den umiddelbare omkostning ved TABLE-MULTI-DELETE(T, k) er $O(1)$, da det er tilstrækkeligt at trække k fra $T.size$. Men sletninger kan føre til omkostninger når tabellen skal sammentrækkes. Vi kan bruge revisormetoden (accounting method), og knytte en kredit på 1 til hvert element i en dynamisk tabel (koster 1 per indsættelse). Kreditten for de slettede elementer er så tilstrækkeligt til at betale for sammentrækningerne.

Tirsdag

1. CLRS 21.2-6. Løsning findes i CLRS løsningsmanual.
2. CLRS 21.2-3. Løsning findes i CLRS løsningsmanual.

Fredag

1. CLRS 21.1-3. FIND-SET kaldes to gange for hver kant, $2|E|$ gange totalt. Vi starter med $|V|$ mængder og slutter med k mængder svarende til sammenhængskomponenterne. Da antal mængder reduceres med én for hvert kald af UNION må der være $|V| - k$ kald.
2. CLRS 10.1-6. En implementation af en kø ved brug af to stakke kan findes i opgaveløsningerne for uge 4. Både ENQUEUE og DEQUEUE bruger amortiseret $O(1)$ tid. Det kan f.eks. ses ved brug af revisormetoden, hvor der knyttes 1 kredit til hvert element i stakken $S1$. Tiden for løkken i DEQUEUE kan så betales af de elementer, der fjernes fra $S1$.