DMA uge 6, løsninger til nogle af tirsdagens opgaver

Nogle af disse løsninger er skrevet ned meget kortfattet. Her må man selv lave "mellemregningerne". 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.

1 (b) Antag at $A[i] = A[j]$, hvor $i < j$. For at vise at COUNTING-SORT er stabil, viser vi at $A[i]$ placeres i $B[i']$ og $A[j]$ placeres i $B[j']$ hvor $i' < j'$. Dette følger af at for-løkken, linje 10 i pseudokoden, CLRS side 195, løber oppefra og ned: Først placeres $A[j]$ i $B[j']$, for $j' = C[A[j]]$, og så trækkes 1 fra $C[A[j]]$. Derved bliver $A[i]$ senere placeret i $B[i']$ for et indeks $i' < j'$.

1 (c) For ethvert tal $i \in \{0, 1, \ldots, k\}$ vil indgangene i A med værdi i blive skrevet i det samme del-array af B som i den oprindelige algoritme, men i omvendt rækkefølge. Derfor vil algoritmen stadig sortere, men den vil ikke være stabil.

2 (b) Vi bruger Lemma 8.4 med $r = \lg n$. Da $b = \lg(n^3 - 1) = 3 \lg n$, fås at køretiden bliver $\Theta((b/r)(n + 2^r)) = \Theta((3 \lg n / \lg n)(n + 2^{\lg n})) = \Theta(n)$.

3 (b) Hvis alle elementerne lander i samme spand, får vi en liste af længde n som skal sorteres. Med insertion sort tager det $\Theta(n^2)$ tid. Modifikation til algoritmen: I stedet for at sortere hver liste med insertion sort, kopierer vi først listens elementer over i et array. Så kan vi bruge merge sort. Da bliver køretiden

$$\Theta\left(\sum_{i=0}^{n-1} n_i \log n_i\right) \leq \Theta\left(\sum_{i=0}^{n-1} n_i \log n\right) = \Theta\left(\left(\sum_{i=0}^{n-1} n_i\right) \log n\right) = \Theta(n \log n).$$