DMA 2021

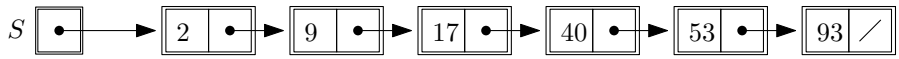
– Ugeopgave 4 –

- Ugeopgaven skal skrives i LaTeX og afleveres som PDF.
- Ugeopgaven skal laves i grupper (som udgangspunkt 2–3 personer), og man skal aflevere som gruppe på Absalon.
- Jeres navne skal fremgå af første side af besvarelsen.
- Alle spørgsmål skal forsøges besvaret for at få opgaven godkendt.

Opgaven

Del A

Lad S være en enkelthægtet liste. Altså har hvert element x i S en nøgleværdi $x.key$ og en pointer $x.next$ til det næste element. Listen S har en pointer $S.head$ til det første element. Vi antager at listen S er sorteret. Med det mener vi at for alle elementer x i S hvor $x.next \neq \text{NIL}$ gælder det at $x.key \leq x.next.key$. Følgende diagram viser sådan en liste S:



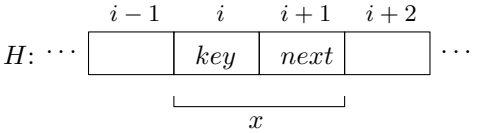
1. Vi er interesserede i at vedligeholde S sådan at listen forbliver sorteret når man indsætter nye elementer. Lav pseudokode til en funktion $\textsc{Insert}(S, x)$, der tager en sorteret liste S og et nyt element x som argumenter. Funktionen skal indsætte x i S, sådan at S stadig er sorteret. Det skal ske på følgende måde: $\textsc{Insert}(S, x)$ gennemløber S fra starten indtil den finder det rigtige sted at indsætte x. Herefter indsættes x. Din pseudokode skal have nummererede linjer.

2. Argumentér for at din funktion i værste fald bruger $\Theta(n)$ tid på indsættelse af et enkelt element, hvor n er antallet af elementer i listen.

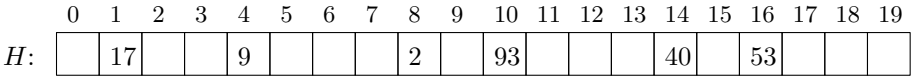
3. Lad listen S være tom til at starte med. Der indsættes herefter n elementer, et ad gangen. Når det sidste element er indsat vil S være en sorteret liste bestående n elementer. Argumentér for at vi på denne måde har sorteret n tal i $O(n^2)$ tid.

4. Hvilken sorteringsalgoritme svarer ovenstående algoritme til?

5. I denne delopgave kigger vi på hvordan en sorteret liste er gemt i hukommelsen H. Hvert element x gemmes som to nabofelter $H[i, i+1]$ i H (tænk på H som et stort array), så key gemmes i $H[i]$ og $next$ gemmes i $H[i+1]$:



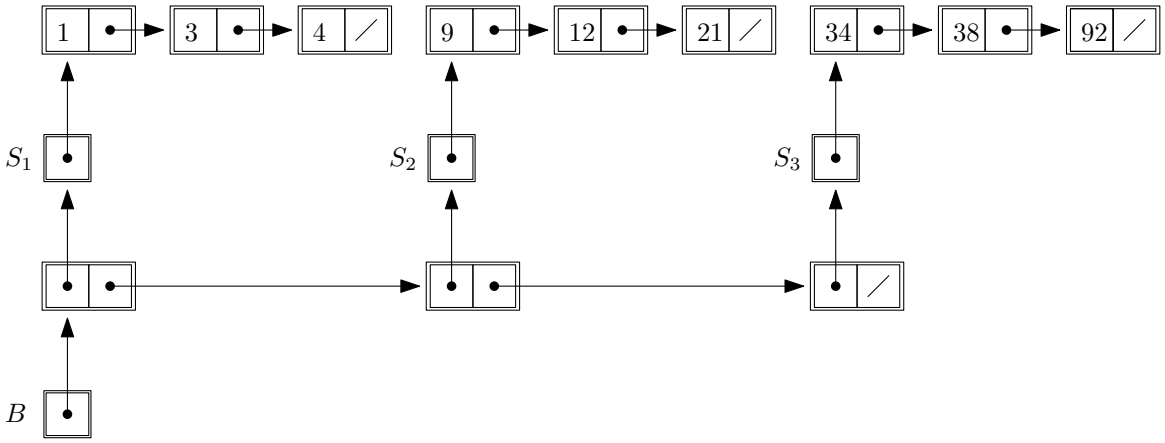
I $next$-feltet står indekset/adressen på feltet i H hvor det næste listeelements nøgleværdi er gemt. Nedenstående figur viser hukommelsen som indeholder elementerne i den liste S, som er afbildet i diagrammet ovenfor, men kun nøgleværdierne er skrevet ind. Skriv $next$-værdierne ind.



Del B

Lad n være et kvadrattal, altså et positivt helt tal således at $k = \sqrt{n}$ er et helt tal. I denne opgave har vi k enkelthægtede lister $S_1, \ldots, S_k$. Hver liste S_i består af k elementer og er sorteret som beskrevet i Del A. Desuden gælder for alle $i \in \{1, \ldots, k-1\}$ at den største nøgleværdi i S_i er højst så stor som den mindste i S_{i+1}, sådan at hvis vi sammensatte listerne $S_1, \ldots, S_k$ til én liste af længde n, ville denne også være sorteret.

Vi har desuden en enkelthægtet liste B af længde k, som er konstrueret så den har følgende egenskab. Lad x være element nummer i i B. Da er $x.key$ en pointer til listen S_i. Følgende diagram viser et eksempel for $n = 9$:



1. Forklar hvorfor $B.head.key.head.key$ er nøgleværdien i det første element i den første liste S_1.
2. Lav pseudokode til en funktion INSERT-FROM-B(B, x) der tager en liste B og et nyt listeelement x som argument. Det antages at nøgleværdien i element nummer i i B er en pointer til en liste S_i, sådan at disse lister $S_1, \ldots, S_k$ er sorterede som beskrevet ovenfor. Det antages også at B indeholder k elementer og at hver liste S_i indeholder k elementer. Funktionen INSERT-FROM-B(B, x) skal bruge tid $O(\sqrt{n})$. Din pseudokode skal have nummererede linjer. *Hint:* Du er velkommen til at kalde din funktion INSERT fra Del A i pseudokoden til INSERT-FROM-B.
3. *(Frivillig – man behøver ikke at lave denne opgave.)* Vis at ovenstående tankegang kan bruges til at sortere n tal i $O(n\sqrt{n})$ tid.