Algoritmer og Datastrukturer (NDAA04010U)
Ugeopgave 1 med svar og forklaringer

Københavns Universitet

2024

1 MergeSort

Antag at MergeSort(A, p, r), implementeret som i CLRS sektion 2.3, bliver kaldt på 21 elementer (dvs. $r-p+1 = 21$). Hvor mange kald bliver der totalt lavet til Merge-Sort? Hvad er antallet af kald generelt, når input har n elementer? Argumentér for dine svar.
Svar: Antal elementer i hvert kald af mergesort, opdelt efter rekursionsniveau, er: 21; 10, 11;5, 5, 5, 6; 2, 3, 2, 3, 2, 3, 3, 3; 1, 1, 1, 2, 1, 1, 1, 2, 1, 1, 1, 2, 1, 2, 1, 2; 1, 1, 1, 1, 1, 1, 1, 1, 1, 1. Antal kald er altså $1+2+4+8+16+10 = 41$. Generelt kan man se at antal indre knuder i et binært træ med n blade altid er $2n-1$ (det kan fx ses ved induktion i n), hvilket er en alternativ måde at komme frem til svaret 41. $\triangle$

2 Rekursionsligninger

Hvilke af disse rekursionsligninger har løsningen $T(n) = \Theta(n^2)$? Antag at $T(n) = 1$ for $n \leq 1$. Vælg ét eller flere korrekte svar og beskriv hvordan du kom frem til dem (både positive og negative svar).

1. $T(n) = 4T(\lfloor n/2 \rfloor) + n \lg n$. ✓
2. $T(n) = 4T(\lfloor n/2 \rfloor) + n^2$.
3. $T(n) = 2T(\lfloor n/4 \rfloor) + n^2$. ✓
4. $T(n) = 9T(\lfloor n/3 \rfloor) + n^3$.
5. $T(n) = T(n-1) + n^2$.
6. $T(n) = T(n-1) + n$. ✓

Svar: De fire første rekursionsligninger kan løses ved hjælp af master theorem, hvor nummer 1 og 3 falder i henholdsvis tilfælde 1 og 3 som giver $T(n) = \Theta(n^2)$. De sidste ligninger kan løses med substitutionsmetoden. I ligning 5 er der n led hvor halvdelen har størrelse mindst $(n/2)^2 = n^2/4$, så $T(n) = \Omega(n^3)$. I ligning 5 er der n led af størrelse højst n, hvor halvdelen har størrelse mindst $n/2$, så $T(n) = \Theta(n^2)$. $\triangle$

3 Del og hersk

Antag at du har en rekursiv algoritme A, og lad n betegne størrelsen på algoritmens input X. Hvis $n = 1$ beregnes resultatet $A(X)$ i konstant tid. For $n > 1$ bruger algoritmen først $O(n \lg n)$ tid på at beregne fire inputs X_1, X_2, X_3, X_4, hver af størrelse $\lfloor n/2 \rfloor$, og $A(X_1), A(X_2), A(X_3), A(X_4)$ beregnes rekursivt. Endelig kombineres de rekursive svar til et output $A(X)$ i tid $O(n^2)$. Skriv en rekursionsligning der beskriver en øvre grænse på køretiden $T(n)$ af A på input X og find en løsning ved hjælp af master theorem. Argumentér for dit svar.

Svar: De fire rekursive delproblemer af størrelse højst $\lfloor n/2 \rfloor$ tager total tid $4T(\lfloor n/2 \rfloor)$. Derudover bruges der totalt tid $O(n^2)$ på at beregne $A(X)$. Fordi vi blot er ude efter en øvre grænse kan vi antage at den ekstra tid er $\Theta(n^2)$, det kan kun gøre tidsforbruge større. Så svaret er $T(n) = 4T(\lfloor n/2 \rfloor) + \Theta(n^2)$. Master theorem (tilfælde 2) giver løsningen $O(n^2 \log n)$. $\triangle$

4 Køretid

Betragt følgende kode i pseudo-kode notationen fra CLRS:

```
LOOPS(n)
r = 0
for i = 1 to n/2
    for j = n/2 − i to n
        r = r + j
return r
```

Vi antager at $n/2$ er et heltal således at den yderste **for**-løkke har $n/2$ iterationer. Hvilke af følgende er gyldige udsagn om køretiden $T(n)$ af LOOPS(n)? (Læg mærke til at udsagnene ikke er uafhængige — hvis der gælder $f(n) = O(n)$ så gælder også $f(n) = O(n \lg n)$, osv.) Vælg ét eller flere korrekte svar og beskriv hvordan du kom frem til dem (både positive og negative svar).

1. $T(n) = O(n)$.
2. $T(n) = O(n \lg n)$.
3. $T(n) = O(n^2)$. ✓
4. $T(n) = \Omega(n)$. ✓

5. $T(n) = \Omega(n \lg n)$. $\checkmark$

6. $T(n) = \Omega(n^2)$. $\checkmark$

Svar: Der er $n/2$ iterationer af den ydre løkke, der hver tager $O(n)$ tid fordi den indre løkke laver mellem $n/2$ og n iterationer. Dvs. køretiden er $\Theta(n^2)$. Derfor er de to første store-O grænser for lave, mens de øvrige grænser gælder. $\triangle$

5 Invarianter

Betragt følgende funktion, i pseudo-kode notationen fra CLRS, hvor input A er en tabel af heltal og n er længden af A.

```
SECOND-SMALLEST(A, n)
m = A[1]
r = ∞
for i = 2 to n
    if A[i] < m
        r = m
        m = A[i]
    elseif A[i] < r
        r = A[i]
return r
```

Hvilke udsagn gyldige løkke-invarianter ved starten af hver iteration i **for**-løkken? Vælg ét eller flere korrekte svar og beskriv hvordan du kom frem til dem (både positive og negative svar).

1. $m = \min(A[1], \ldots, A[i-1])$. $\checkmark$

2. $m = \min(A[1], \ldots, A[n])$.

3. $r \geq m$. $\checkmark$

4. $m \geq 0$.

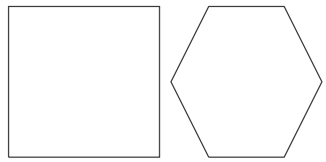
5. Mindst 1 element i $A[1], \ldots, A[i-1]$ er mindre end eller lig med r. $\checkmark$

6. Højst 2 elementer i $A[1], \ldots, A[i-1]$ er mindre end eller lig med r.

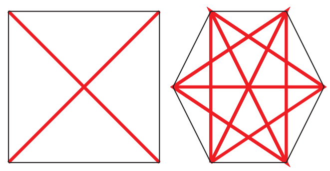
Svar: Udsagn 1, 3 og 5 gælder for $i = 2$ pga. initialiseringen af m og r, og kan ses at gælde efter hver iteration ved induktion. Udsagn 2 gælder ikke hvis minimum i A ikke er lig med $A[1]$. Udsagn 4 gælder ikke hvis $A[1] < 0$. Udsagn 6 gælder ikke hvis der er 3 værdier i A, der har den samme, mindste værdi. $\triangle$

6 Induktionsbeviser

I denne opgave betragter vi regulære n-kanter som disse eksempler ($n = 4$ og $n = 6$):



Vi er interesserede i antal *diagonaler*, dvs. linjer der forbinder to hjørner i n-kanterne og er helt indeholdt i figuren. En 3-kant har ingen diagonaler da alle hjørner er naboer, mens vi for $n = 4$ og $n = 6$ har henholdsvis 2 og 9 diagonaler.



En diagonal deler en n-kant i en n_1-kant og en n_2 kant med en fælles kant og hvor $n_1 + n_2 = n + 2$. For eksempel deler 4-kantens diagonal den i to 3-kanter.

a) Bevis ved induktion at antal diagonaler d_n i en n-kant er lig med $n(n-3)/2$ for $n \geq 3$. Der lægges vægt på, at argumentationen er klar og koncis.

Svar: For $n = 3$ har vi $d_n = 0 = 3(3-3)/2$, hvilket etablerer basistilfældet. Antag nu at $n > 3$, at udsagnet er korrekt for n'-kanter med $n' < n$, og betragt en n-kant. Tag tre hjørner v_1, v_2, v_3 der ligger ved siden af hinanden, dvs. hvor v_1 og v_3 er forbundet til v_2. Fra v_2 kan der laves diagonaler til hver af de andre $n-3$ hjørner og desuden kan der laves en diagonal mellem v_1 og v_3. De resterende diagonaler ligger i den $(n-1)$-kant, der fås ved at udelade v_2 (og forbinde v_1 og v_3 direkte). Ifølge induktionshypotesen har vi $d_{n-1} = (n-1)(n-4)/2$ diagonaler i denne del, dvs.

$$d_n = n - 3 + 1 + d_{n-1} = n - 2 + (n-1)(n-4)/2 = n(n-3)/2 \ .$$

△

7 Korrekthed og køretid

I denne opgave bruger vi matrix og vektor-notation, se evt. CLRS appendix D.1. Fibonaccitallene $F_0, F_1, F_2, \ldots$ er defineret ved $F_0 = 0$, $F_1 = 1$, og $F_n = F_{n-1} + F_{n-2}$ for $n > 1$. Betragt nu matricen A og søjlevektorer på formen $x = (x_1, x_2)^T$.

$$A = \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix} \qquad x = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$$

a) Argumentér for at for $x = (F_{n-2}, F_{n-1})^T$ opfylder matrix-vektor produktet ligheden $Ax = (F_{n-1}, F_n)^T$ for $n > 1$.

Svar: Per definition af matrix-vektor produkter har vi:

$$A = \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} F_{n-2} \\ F_{n-1} \end{pmatrix} = \begin{pmatrix} F_{n-1} \\ F_{n-2} + F_{n-1} \end{pmatrix} = \begin{pmatrix} F_{n-1} \\ F_n \end{pmatrix},$$

hvor den sidste lighed bruger definitionen af F_n. $\triangle$

b) Bevis at for $x = (0, 1)^T$ gælder ligheden $A^{n-1}x = (F_{n-1}, F_n)^T$ for $n \geq 1$.

Svar: Bevis ved induktion i n. Basistilfældet $n = 1$ gælder da $A^{n-1}x = x = (F_0, F_1)^T$. Antag som induktionshypotese at $A^{i-1}x = (F_{i-1}, F_i)^T$ for $1 \leq i < n$. Da har vi

$$A^{n-1}x = A(A^{n-2}x) = A(F_{n-2}, F_{n-1})^T = (F_{n-1}, F_n)^T,$$

hvor lighed nummer 2 bruger induktionshypotesen og lighed nummer 3 bruger del a). $\triangle$

c) Argumentér for at A^{n-1}, og dermed også $A^{n-1}x$, kan udregnes i tid $O(\lg n)$, hvis vi antager at multiplikation, addition og subtraktion tager konstant tid. (Start evt. med at se på tilfældet hvor $n - 1$ er en potens af 2.)

Svar: Vi har følgende rekursionsligning:

$$A^i = \begin{cases} I & \text{hvis } i = 0 \\ (A^{i/2})^2 & \text{hvis } i > 0 \text{ er lige} \\ A(A^{(i-1)/2})^2 & \text{hvis } i \text{ er ulige} \end{cases}$$

Da A^{n-1} er af konstant størrelse betyder det at A^{n-1} kan udregnes i konstant tid ud fra enten $A^{(n-1)/2}$ eller $A^{(n-2)/2}$. Det giver rekursionsdybde højst $\lg(n)$, og dermed tid $O(\lg n)$. $\triangle$

d) Argumentér for at F_n kan udregnes i tid $O(\lg n)$ hvis vi antager at aritmetiske operationer tager konstant tid.

Svar: Følger af b) og c). $\triangle$