Ugeseddel: Introduktion

Philip Bille

Om denne uge

Litteratur *Introduction to Algorithms*, Cormen, Rivest, Leisersons og Stein (CLRS): Kap 1.

Opgaver

1 Find toppunkter Lad $A = [2, 1, 3, 7, 3, 11, 1, 5, 7, 10]$ være en tabel. Løs følgende opgaver.

1.1 $[o]$ Angiv alle toppunkter i A.

1.2 $[o]$ Angiv hvilke toppunkter de to lineærtidsalgoritmer finder.

1.3 Angiv sekvensen af rekursive kald, som den rekursive algoritme producerer. Antag først at rekursionen fortsætter med venstre halvdel af tabellen hvis der er mulighed for at gå begge veje. Angiv derefter alle mulige sekvenser af rekursive kald der kan opnås ved frit valg når man kan gå begge veje.

2 Lavpunkter Giv en præcis definition af *lavpunktsproblemet*.

3 Algoritmer og datastrukturer

3.1 CLRS $[o]$ 1.1-1.

3.2 CLRS $[o]$ 1.1-2.

3.3 CLRS 1.1-3.

3.4 CLRS 1.1-5.

3.5 CLRS 1.2-1.

3.6 CLRS 1.2-3.

4 Egenskaber for toppunkter Lad A være en tabel af længde $n \geq 1$. Løs følgende opgaver.

4.1 Bevis at der altid er mindst et toppunkt i A.

4.2 Hvad er det maximale antal toppunkter, der kan være i A?

5 Toppunkter Løs følgende opgaver.

5.1 $[\dagger]$ Implementer og afprøv en af de to lineærtidsalgoritmer til at finde toppunkter.

5.2 $[\dagger]$ Implementer den rekursive algoritme til at finde toppunkter (vær opmærksom på ikke at komme til at gå ud over grænserne på tabellen).

5.3 Beskriv hvordan værstefaldsinput til hver af de 3 toppunktsalgoritmer ser ud.

5.4 $[D{*}\dagger]$ Skriv pseudokode for en iterativ version af den rekursive algoritme til at finde toppunkter. Implementer og afprøv den.

5.5 $[C]$ Bevis at den rekursive løsning altid finder et toppunkt. *Hint:* Definer en passende invariant som er tilfredsstillet ved hvert rekursivt kald og benyt induktion.

6 Køretider Løs følgende opgaver.

6.1 CLRS 1.2-2.

6.2 CLRS 1.1.

7 2D toppunkter Lad M være $n \times n$ matrix (2D-tabel). En indgang $M[i,j]$ er et *toppunkt* hvis det ikke er mindre end dets naboer i retning N, Ø, S og V (dvs. $M[i][j] \geq M[i-1][j]$, $M[i][j] \geq M[i][j-1]$, $M[i][j] \geq M[i+1][j]$ og $M[i][j] \geq M[i][j+1]$). Vi er interesseret i effektive algoritmer til at finde et toppunkt i A. Løs følgende opgaver.

7.1 Giv en algoritme der tager $\Theta(n^2)$ tid.

7.2 [$*$] Giv en algoritme der tager $\Theta(n \log n)$ tid. *Hint:* Start med at finde det maksimale tal i den miderste søjle og benyt det til at lave en rekursion.

7.3 [$**$] Giv en algoritme der tager $\Theta(n)$ tid. *Hint:* Konstruer en rekursion der inddeler M i 4 kvadranter.

O Obligatorisk afleveringsopgave: Sjov med tabeller Lad A være en tabel af heltal af længde n. Kig på nedenstående pseudokode og løs følgende opgaver.

```
TabelSjov(A, n)
for i = 0 to n − 1 do
   for j = 0 to n − 1 do
      for k = 0 to n − 1 do
         if A[i] + A[j] + A[k] = 0 then
            return true
         end if
      end for
   end for
end for
return false
```

O.1 Forklar kort og præcist hvad TabelSjov udregner.

O.2 Analyser køretiden af TabelSjov på en tabel af længden n.

O.3 Skriv et program, der implementerer TabelSjov algoritmen.

O.4 Skriv et program, der afprøver din implementation på et par eksempler. Afprøv dit program og kommenter kort på resultaterne.