

Metoder til algoritmedesign



Algoritmedesign er en **kreativ** proces. Der findes ingen generel mekanisk metode (algoritme) til design af en algoritme for et forelagt problem

Derimod findes en række metoder, eller rettere “regler for tænkning”, som ofte fører til korrekte og effektive algoritmer

Nogle af disse metoder er baseret på matematisk bevisførelse. Dermed “indbygges” korrekthed i algoritmerne

Andre har mere karakter af gode råd

1

Problemløsningsteknik

(“How to solve it”, Polya - 1944)



1. Forstå problemet

Hvad er det, der søges? Hvad er givet? Tegn en figur. Indfør passende betegnelser

2. Læg en plan

Har du set dette problem eller tilsvarende tidligere? Kan problemet omformuleres? Kan du løse en del af problemet?

3. Gennemfør planen

Kontroller hvert trin. Kan du bevise hvert trin?

4. Se tilbage

Kan du kontrollere resultatet? Kan du opnå resultatet på anden måde? Kan du anvende metoden på et andet problem?

2

Berømthedsproblemet



Blandt N personer defineres en **berømthed** som en person, som alle kender, men som ikke kender nogen andre

Identificer berømtheden, hvis denne eksisterer, ved kun at stille spørgsmål af formen: “Undskyld, kender du den person derhenne?”

Alle personer, også berømtheden, antages at svare korrekt

Antallet af spørgsmål ønskes minimeret

3

1. Forstå problemet.

Hvad er det, der søges?

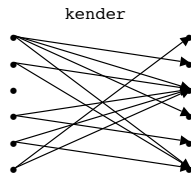
En person blandt N personer, som opfylder betingelsen for at være en berømthed. Hvis der ikke findes nogen berømthed, skal dette meddeles. Antag at personerne nummereres fra 1 til N . Så er det søgte: nummeret på en person. Fravær af berømthed kan angives med et ikke-eksisterende personnummer, f.eks. 0

Hvad er givet?

Et 2-dimensionalt boolean array (logisk matrix), **kender**, der for hvert ordnet par af personer (a,b) , $1 \leq a \leq N$, $1 \leq b \leq N$, angiver om a kender b (**kender**[a][b] er **true**, hvis og kun hvis a kender b)

4

Tegn en figur.



En berømthed svarer til en knude, hvor antallet af udgående pile er 0, og antallet af indgående pile er $N-1$

En berømthed er karakteriseret ved, at alle elementer i den tilsvarende række i `kender` er `false`, mens alle elementer i den tilsvarende søjle er `true` (idet der ses bort fra diagonalelementerne)

Indfør passende betegnelser.

Er gjort

5

2. Læg en plan

Har du set dette problem eller tilsvarende tidligere?

Nej, men jeg kan lave en meget simpel algoritme til løsning af problemet.

Undersøg for hver person, om `kender` opfylder betingelsen for, at personen er en berømthed:

```
berømthed = 0;
for (a = 1; a <= N; a++) {
  falseAntal = trueAntal = 0;
  for (b = 1; b <= N; b++)
    if (b != a && !kender[a][b])
      falseAntal++;
  for (b = 1; b <= N; b++)
    if (b != a && kender[b][a])
      trueAntal++;
  if (falseAntal == N-1 && trueAntal == N-1)
    berømthed = a;
}
```

Simplet, men ikke særlig effektiv. Berømtheden findes ved at stille $N*2*(N-1)$ spørgsmål. Kan denne løsning forbedres?

6

Ja, ved at undgå tællerne `falseCount` og `trueCount` og forlade løkkerne, så snart det er muligt

```
berømthed = 0;
spørgen:
for (a = 1; a <= N; a++) {
  for (b = 1; b <= N; b++)
    if (b != a &&
        (kender[a][b] || !kender[b][a]))
      continue spørgen;
  berømthed = a;
  break;
}
```

Hvis der eksisterer en berømthed, stilles **gennemsnitligt** færre spørgsmål, men i **værste** tilfælde stilles $N*2*(N-1)$ spørgsmål

Kan vi mon gøre det bedre?

7

Ja. Vi kan eliminere nogle af spørgsmålene. Vi kan nemlig risikere at måtte stille et spørgsmål to gange. Hvis $b < a$, kan spørgsmålet "`kender[a][b]`?" være besvaret tidligere, nemlig i form af svaret på spørgsmålet "`!kender[b][a]`?"

Vi når frem til følgende algoritme.

```
berømthed = 0;
spørgen:
for (a = 1; a <= N; a++) {
  if (berømthed != 0 &&
      (kender[berømthed][a] ||
       !kender[a][berømthed]))
    berømthed = 0;
  if (berømthed == 0) {
    for (b = 1; b < a; b++)
      if (kender[a][b] || !kender[b][a])
        continue spørgen;
    berømthed = a;
  }
}
```

I **værste** tilfælde stilles $N*(N+1)$ spørgsmål. $[= \sum_{a=1}^N (2 + 2(a-1))]$

8

Kan problemet omformuleres?

Det kan være svært at identificere en berømtthed. Det er måske lettere at identificere personer, som ikke er berømttheder (i det mindste er de i flertal). Problemet kan omformuleres: Find $N-1$ personer, der ikke er en berømtthed, og påvis, at den tiloversblevne er en berømtthed

Hvis $kender[a][b]$, $a \neq b$, kan det udelukkes, at a er en berømtthed. I modsat fald kan det udelukkes, at b er en berømtthed. En af dem kan elimineres med ét spørgsmål. Hvert spørgsmål reducerer problemet til et tilsvarende, men med en person færre

9

```
for (a = 1, b = 2, n = 3; n <= N + 1; n++)  
    if (kender[a][b]) a = n; else b = n;  
if (a == N + 1) a = b;
```

```
for (b = 1; b <= N; b++)  
    if (b != a &&  
        (kender[a][b] || !kender[b][a]))  
        break;  
berømtthed = b > N ? a : 0;
```

Antallet af spørgsmål er nu reduceret til $3*(N-1)$ i værste tilfælde

10

3. Gennemfør planen

Kontroller hvert trin. Kan du bevise hvert trin?

Planen er gennemført. Et *formelt* bevis for algoritmens korrekthed ville være ideelt, men i det mindste er der argumenteret for dens korrekthed.

4. Se tilbage

Kan du kontrollere resultatet?

Ja, f.eks. empirisk ved systematisk programafprøvning. Men spørgsmålet er, om effektiviteten er den bedst opnåelige. Faktisk kan der på snedig vis spares $\log_2 N$ spørgsmål.

11