Forén og find

Eva Rotenberg*

Materialer CLRS: kap. 21.1–21.4; *Algorithms*, 4. udgave, Sedgewick og Wayne: kap 1.5.

Opgaver

1 Håndkøring af forén og find

Kig på følgende sekvens af operationer: INIT(7), UNION(3, 4), UNION(5, 0), UNION(4, 5), UNION(4, 3), UNION(0, 1), UNION(2, 6), UNION(0, 4) og UNION(6, 0).

1.1 Håndkør sekvensen med hurtig find. Vis indholdet af *id* arrayet efter hvert skridt. Antag at UNION(i, j) operationen altid opdaterer id for mængden angivet ved i.

1.2 Håndkør sekvensen med hurtig forening. Vis træerne efter hvert skridt. Antag at UNION(i, j) altid sætter roden af træet angivet ved i til at være barn af roden af træet angivet ved j.

1.3 Håndkør sekvensen med vægtet forening. Vis træerne efter hvert skridt. Antag at UNION(i, j) sætter roden af træet angivet ved i til at være barn af roden af træet angivet ved j når træerne har samme størrelse.

1.4 Vis resultat af stikompression efter en FIND(x) operation, hvor x er hhv. et blad, en intern knude af dybde 1 og en intern knude af højde 1, i et af træerne fra øvelserne ovenfor.

1.5 Giv en sekvens af operationer der fører til et træ af maksimal dybde i hurtig forening.

1.6 Giv en sekvens af operationer der fører til et træ af maksimal dybde i vægtet forening.

1.7 Skriv pseudokode for en algoritme til stikompression. *Hint:* Løb stien igennem to gange.

2 Alternativ algoritme til forening i datastrukturen med hurtig find

En ven foreslår følgende intuitive variant af UNION til datastrukturen med hurtig find. Virker den?

UNION(i, j)
```
if FIND(i) ≠ FIND(j) then
  for k = 0 to n − 1 do
    if id[k] == id[i] then
      id[k] = id[j]
    end if
  end for
end if
```

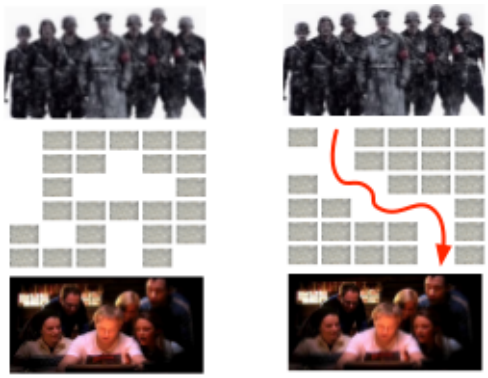
3 Dynamiske sammenhængskomponenter og grafsøgning

Med grafsøgning (såsom dybdesøgning eller breddesøgning) kan vi finde sammenhængskomponenterne i en graf. Giv en simpel løsning til dynamiske sammenhængskomponenter vha. grafsøgning og sammenlign kompleksiteten med løsningerne til forén og find.

*baseret på materiale af Billes&Gørtz

4 Zombieinvasion

I den post-apokalyptiske zombieverden har du og en lille gruppe overlevende forskanset jer i en mindre beboelse. Det eneste, der afholder den glubske hær af zombier i at komme ind og spise dig og dine venner, er solidt fæstningværk. Fæstningsværket består af et $k \times k$ gitter af mure. Illustreret her med et 6×6 gitter af mure.[1]



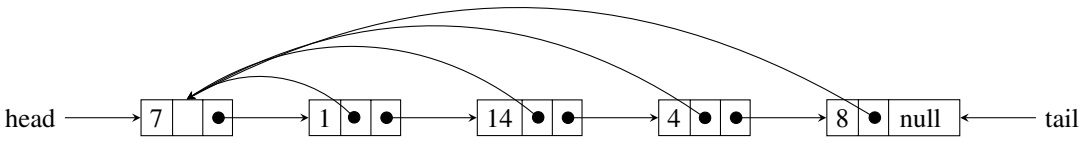
I toppen af gitteret venter zombierne på at komme ind og i bunden er beboelsen. Murerne er desværre skrøbelige, og falder derfor sammen med jævnlige mellemrum. Hvis en sti af mure mellem toppen og bunden af gitteret et faldet sammen kan zombierne komme ind. For at kunne nå at evakure vil du derfor gerne kunne holde styr på om der er en sti igennem fæstningsværket eller ej. Giv en datastruktur, der effektivt kan holde styr på, om der er en sti, efterhånden som mure falder sammen.

5 [∗] Rekursiv stikompression

Skriv pseudokode for en rekursiv algoritme til stikompression. *Hint:* Det kan gøres med meget lidt kode.

6 Forén og find med hægtede lister og vægte

Vi ønsker at implementere en variant af hurtig find med hægtede lister på følgende måde. Hver mængde er repræsenteret ved en enkelt hægtet liste. Repræsentanten for en mængde er det første element i listen og hvert element i listen har en pointer til repræsentanten. Derudover vedligeholder vi en pointer til halen af listen. F. eks. kan datastrukturen for mængden $\{1, 4, 7, 8, 14\}$ med repræsentant 7 se sådan ud:



6.1 Vis hvordan man i denne repræsentation kan implementere Init(n) i $O(n)$ tid, Find(i) i $O(1)$ tid og Union(i, j) i $O(|S(i)|)$ tid, hvor $S(i)$ er mængden der indeholder i. *Hint:* Brug også et array at pointere ved siden af listerne.

6.2 Vis hvordan man kan udvide løsningen således at Init og Find kører i samme tid som i opg.7.1, men tiden for Union(i, j) bliver $O(\min\{|S(i)|, |S(j)|\})$. *Hint:* Vedligehold lidt ekstra information.

6.3 [∗] Vis at løsningen fra delopgave 2 giver at køretiden for p Find og m Union operationer på n elementer er $O(p + m \log n)$.

[1] Opgave af Philip Bille og billeder fra *Død snø, 2009*