

Exjobbsförslag: Hyperklickar i hypergrafer

Fredrik Kuivinen, freku@ida.liu.se

En k -likformig hypergraf H består av en mängd noder V och en mängd hyperkanter E . Varje *hyperkant* är en mängd av k stycken noder. (En vanlig graf är alltså en 2-likformig hypergraf.) En *hyperklick* i H består av en delmängd C av noderna så att för varje delmängd X av C med storlek k så är X en kant i H . En hyperklick är alltså en mängd noder sådan att alla möjliga kanter mellan noderna i klicken ingår i grafen. Om $k = 2$ (hypergrafen är alltså en graf) så sammanfaller hyperklickarna med de vanliga klickarna i grafen.

I en graf utgör en triangel en klick av storlek 3. Det är känt att givet en graf med n noder så är det möjligt att avgöra om grafen har en triangel i $O(n^\omega)$ tid. Här är ω ett reellt tal sådant att det går att multiplicera två stycken $n \times n$ matriser i tid $O(n^\omega)$. Man vet att $\omega < 2.38$.

Det här projektet går ut på att försöka hitta en algoritm som kan avgöra om en k -likformig hypergraf har en hyperklick av storlek $k+1$ i tid $O(n^{k+1-\epsilon})$ för något $\epsilon > 0$. Enligt resultatet ovan har man alltså en sådan algoritm för fallet $k = 2$.

En observation är att det finns en enkel algoritm med körtid $O(n^{k+1})$: gå igenom alla möjliga delmängder av V av storlek $k+1$ och testa ifall de utgör en hyperklick. Det finns $O(n^{k+1})$ sådana delmängder och att testa om en specifik delmängd utgör en hyperklick tar konstant tid. Algoritmen går alltså inom den angivna tidsgränsen.

Frågan är om man kan hitta en snabbare algoritm. I fallet med 3-likformiga hypergrafer blir frågan om det finns en algoritm med körtid $O(n^{4-\epsilon})$ för något $\epsilon > 0$ (t ex finns det en algoritm med körtid $O(n^{3.999})$?)

Projektet kan bedrivas på (minst) två olika sätt.

- I den första metoden har man ett mer teoretisk perspektiv där man sitter och funderar och klurar ut en möjlig algoritm. Troligen kommer arbetet börja med att sätta sig in i teorin bakom snabba matris-multiplikationsalgoritmer.
- Den andra varianten är mer praktisk och exjobbet blir ett programmeringsprojekt. Den här metoden grundar sig på att en möjlig klass av algoritmer utgör lösningen till ett visst icke-linjärt ekvationssystem där varje ekvation består av ett polynom. Problemet blir då att

skriva ett program som kan hitta lösningar till dessa speciala system av polynomekvationer som går så snabbt som möjligt. Om detta tillvägagångssätt väljs kan det bli aktuellt att köra programmet på någon av NSCs superdatorer.

Låter det här intressant? Kontakta isåfall författaren.