Teoriuppgifter labb 3

Tim Olsson

10 oktober 2015

Uppgift 1

Om algoritmen implementeras med en grannmatris behöver man i dess bfs, för varje nod v, iterera genom varje annan nod för att se om de är grannar. Denna operation går i O(V^2). Om man däremot väljer att implementera den med hjälp av en grannlista slipper man kolla om andra noder är grannar. Vid detta fall har man ju redan en lista med alla grannar för varje nod. Därför sker denna operation i $O(V + E)$.

Om $m = O(n)$ går grannlistans bfs i $O(n)$ och grannmatrisens i $O(n^2)$. While-slingan kommer att köras som flest n gånger då det är det maximala flödet vi kan få ut (varje kant kan bara ha max 1 i flöde) och detta multiplicerat med bfs ger oss tidskomplexiteten då man kan ignorera konstanter och initialiseringen. Den bästa lösningen blir därför att använda sig utav grannlistan då $m = O(n)$, detta därför att tidskomplexiteten blir $O(n^2)$ mot $O(n^3)$ om man använder sig utav matrisimplementationen.

Uppgift 2

Det Kalle missar är att stigarna riktas om när de passeras och i värsta fall leder detta till att man kommer att behöva gå sick-sack genom hela den bipartita grafen. Lägger man till en en ny källa s och en ny slutnod t får man två till kanter. Stigen kommer att gå sick-sack från en vänsternod till en högernod så långt den kan gå utan att gå till en nod två gånger
$s \to v \to h \to ... \to v \to h \to t$.
Den hela stigen inklusive s och t innehåller då $1 + 2 * min(|v|, |h|)$ antal noder.

Uppgift 3

Då algoritmen går igenom grafen lägger den till och tar bort kapaciteter från tidigare tal på kapaciteter. Eftersom kapaciteterna är heltal så kommer flödena då även att bli det. De enda operationerna som utförs är addition och subtraktion på heltal vilket då även resulterar i heltal.