

Name: \_\_\_\_\_

1. Jämför tidskomplexiteten för algoritmen då grafen implementeras som en grannmatris och då den implementeras med grannlistor. (För att satsen  $f[v,u] := -f[u,v]$  ska kunna implementeras effektivt måste grannlisteimplementationen utökas så att varje kant har en pekare till den omvända kanten.)

Uttryck tidskomplexiteten i  $n$  och  $m$  där  $n$  är totala antalet hörn och  $m$  antalet kanter i den bipartita grafen. Välj sedan den implementation som är snabbast då  $m = O(n)$ , alltså då grafen är gles.

Eftersom  $m \in O(n)$  så kommer vi aldrig ha särskilt många kanter.

**Grannmatris:**

*for* varje kant: tar  $n^2$  operationer ty vi måste gå igenom alla platser i vår  $n \cdot n$  matris.

**BFS:** tar  $n^2$  eftersom vi för varje nod  $n$  måste undersöka  $n$  noder.

*while* det finns stig  $p$  från  $s$  till  $t$ : tar  $O(n)$  eftersom det i värsta fall finns  $n - 1$  kanter från en nod.

*for* varje kant in  $p$ : tar  $m$  operationer eftersom vi har  $m$  kanter.

**Grannlista:**

*for* varje kant:  $n + 2 \cdot m$  operationer, eftersom vi för varje nod måste gå igenom alla dess kanter. Och varje kant finns två gånger för varje granne.

**BFS:**  $n + 2 \cdot m$  operationer av samma anledning som ovan.

*while* det finns stig  $p$  från  $s$  till  $t$ : tar  $O(n)$  eftersom det i värsta fall finns  $n - 1$  kanter från en nod.

*for* varje kant in  $p$ : För att hitta alla kanter måste vi gå igenom  $2 \cdot m$ .

Vi torde således använda en grannlista. Eftersom vi vet att antalet kanter  $m \leq k \cdot n$ , alltså en väldigt gles graf, kommer grannlistorna aldrig att bli särskilt långa. Vi får tidskomplexiteten  $O(n^3)$  för grannlista och  $O(n^4)$  för grannmatris.

2. Kalle menar att om vi börjar med en bipartit graf  $G$  och gör om den till en flödesgraf  $H$  med ny källa  $s$  och nytt utlopp  $t$  så kommer avståndet från  $s$  till  $t$  att vara 3. Kalle tycker därför att BFS-steget alltid kommer att hitta en stig av längd 3 i restflödesgrafen (om det finns någon stig). Det första påståendet är sant, men inte det andra. Varför har stigarna som BFS hittar i restflödesgrafen inte nödvändigtvis längd 3? Hur långa kan de bli?

Ett uppenbart faktum är att den längsta stigen finns i  $O(m)$ , eftersom den längsta stigen går igenom alla kanter. En bättre uppskattning är dock  $\min(X, Y) \cdot 2 + 1$ , där  $X$  och  $Y$  är mängden noder på var sin sida i den bipartita grafen.

3. Anledningen till att bipartit matchning kan reduceras till flöde är att en lösning till flödesproblemet kan tolkas som en lösning till matchningsproblemet. Detta gäller bara om det flöde som algoritmen ger är ett heltalsflöde (flödet i varje kant är ett heltal), vilket i detta fall innebär att flödet längs en kant antingen är 0 eller

**1. Som tur är så är det på det sättet. Bevisa att Ford-Fulkerson alltid genererar heltalsflöden om kantkapaciteterna är heltal!**

Det skapas aldrig något decimaltal i algoritmen eftersom vi aldrig utför någon delningsoperation. Så om vi har heltal som kapacitet kommer vi också få heltal som flöde.