

TDDI16 – Föreläsning 7

Fler grafalgoritmer – Grafer överallt!

Filip Strömbäck

Planering

Vecka	Fö	Lab
36	Komplexitet, Linjära strukturer	----
37	Träd, AVL-träd	1---
38	Hashning, meet-in-the-middle	12--
39	Grafer, grafteraversering och kortaste vägen	-2--
40	-	--3-
41	Sortering	--34
42	Repetition	---4

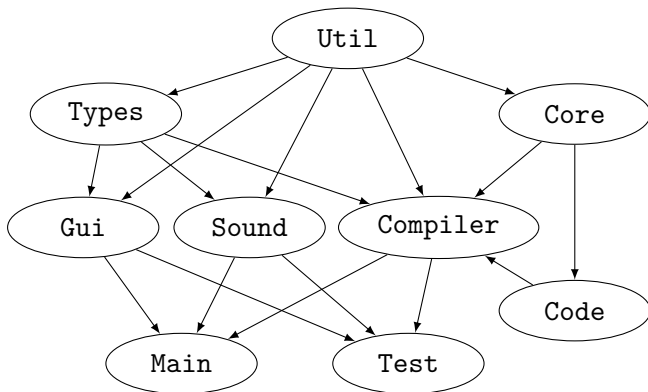
- 1 Topologisk sortering
- 2 Minsta uppspannande träd
- 3 Maxflow
- 4 Sammanfattning

Problem

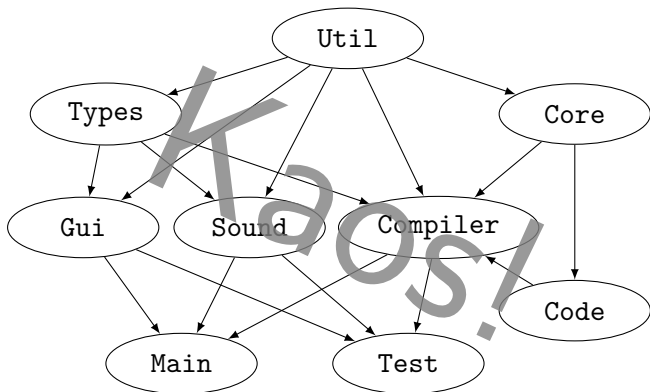
Du ska kompilera ett stort mjukvaruprojekt som består av ett antal delprojekt. Dessa projekt beror på varandra på olika sätt, oftast på ett sådant sätt att det som produceras av ett projekt används för att bygga andra projekt.

I och med att projektet har vuxit sig så stort har du blivit trött på att hålla koll på alla beroenden själv, så du vill bygga ett program som håller koll på beroendena åt dig, och bygger projekten i rätt ordning utifrån det.

Problem



Problem



lakttagelser:

Det här är ju en graf!

Vi vill hitta en sekvens av noder så att:

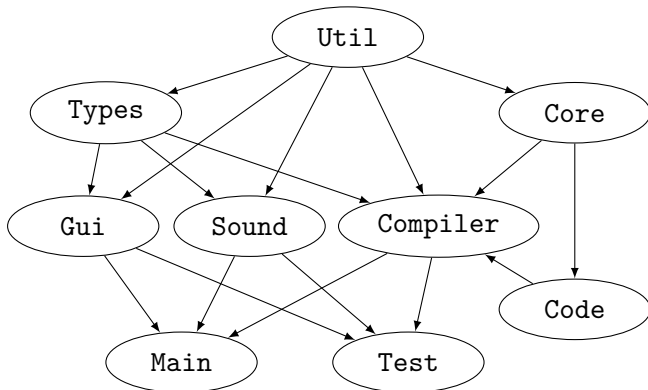
- Om det finns en båge från a till b så ska a vara före b
 - Alla noder finns med exakt en gång
- ⇒ Grafen får alltså inte innehålla cykler
- ⇒ Finns ofta flera sekvenser

Kallas **topologisk sortering**

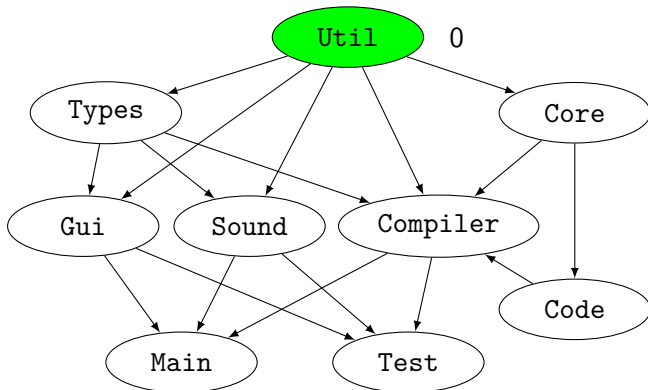
Topologisk sortering – Idé:

1. Hitta alla noder utan inkommande bågar ($\text{indegree} = 0$), lägg dem på en kö, Q
2. Välj den första noden i kön, lägg den sist i resultatsekvensen, R
3. För varje utåtgående båge: minska nodens grad med 1 och lägg på Q om den blev 0
4. Repetera från steg 2 tills Q är tom

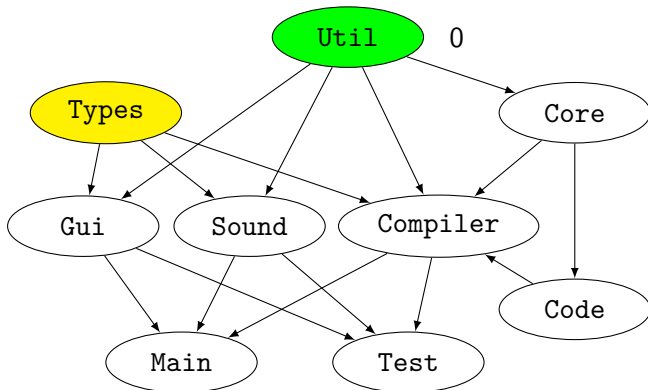
Topologisk sortering – Exempel



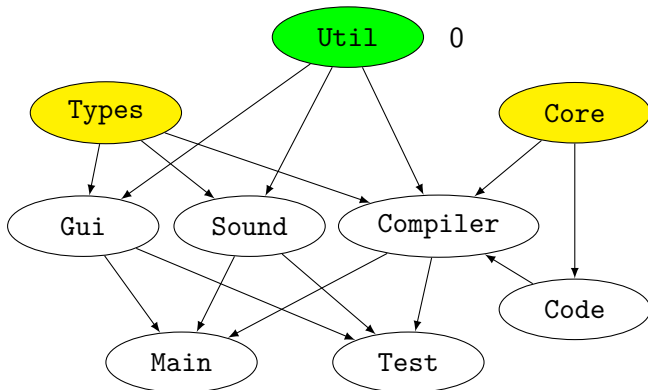
Topologisk sortering – Exempel



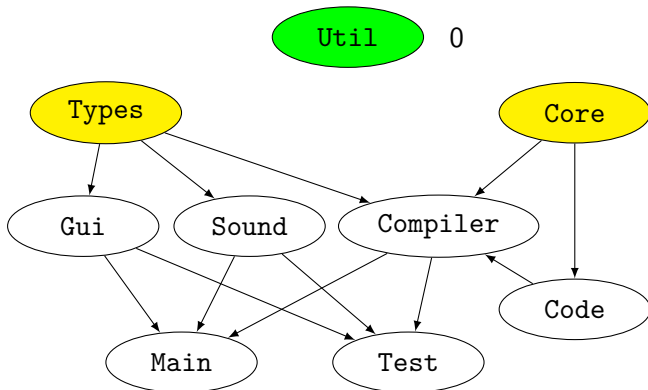
Topologisk sortering – Exempel



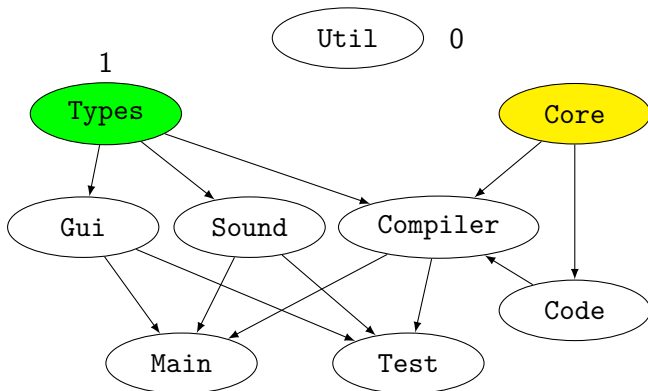
Topologisk sortering – Exempel



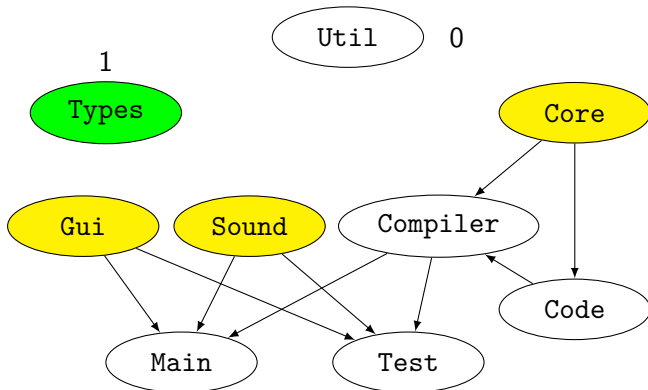
Topologisk sortering – Exempel



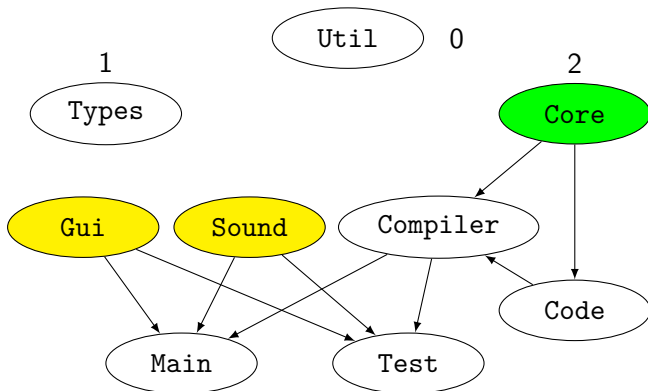
Topologisk sortering – Exempel



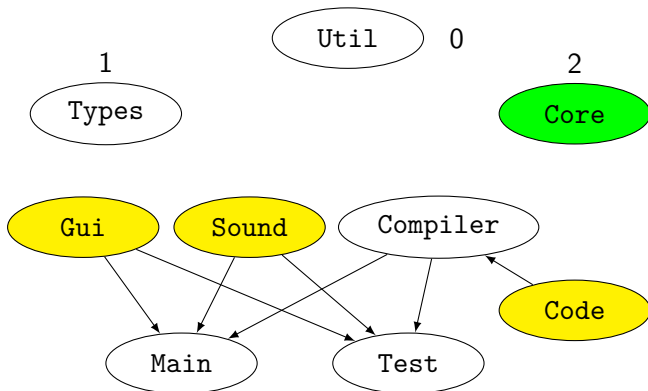
Topologisk sortering – Exempel



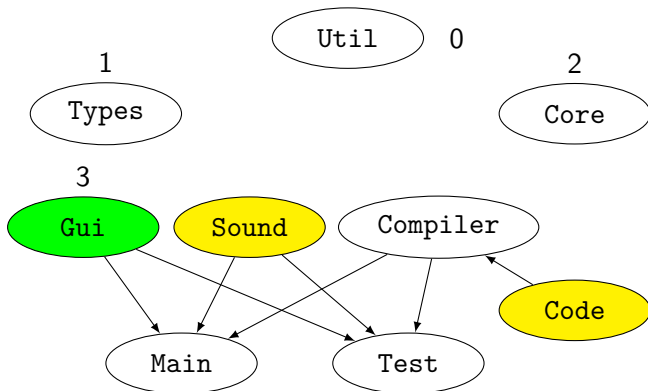
Topologisk sortering – Exempel



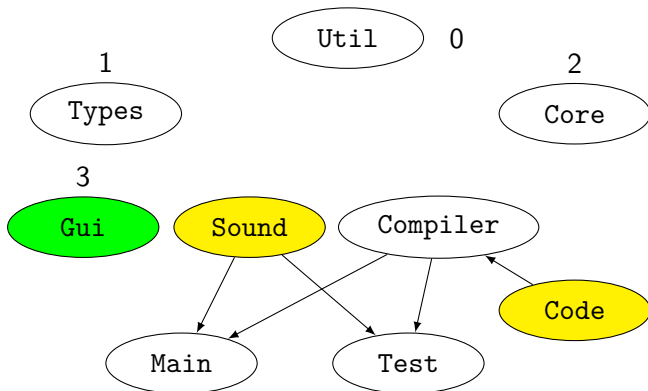
Topologisk sortering – Exempel



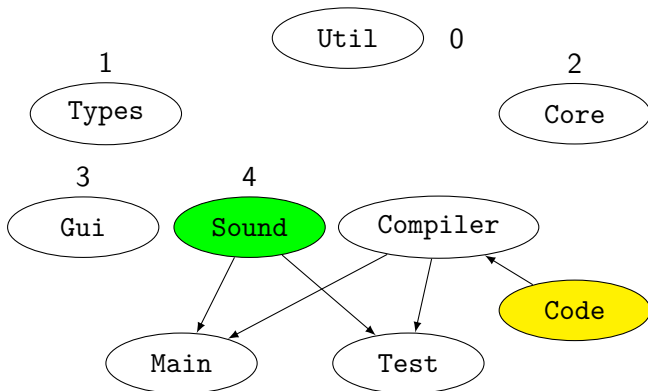
Topologisk sortering – Exempel



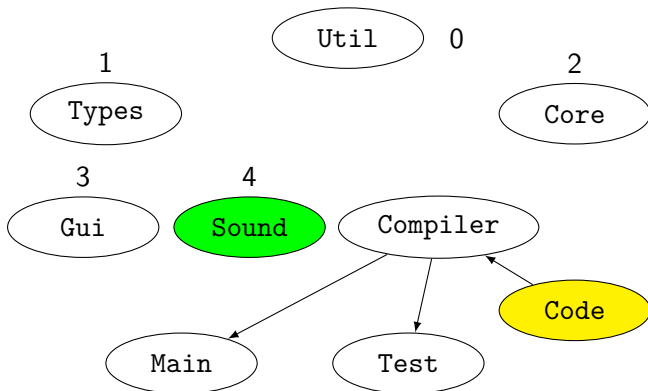
Topologisk sortering – Exempel



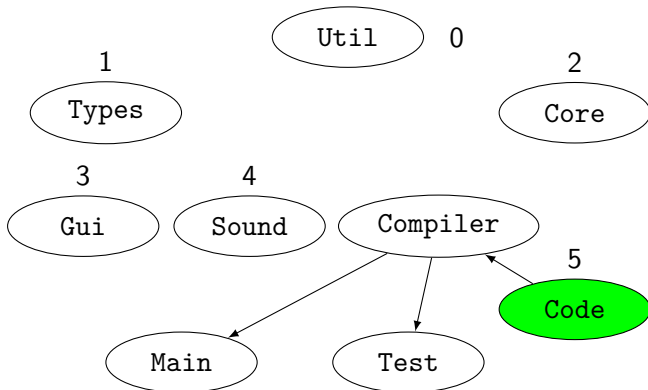
Topologisk sortering – Exempel



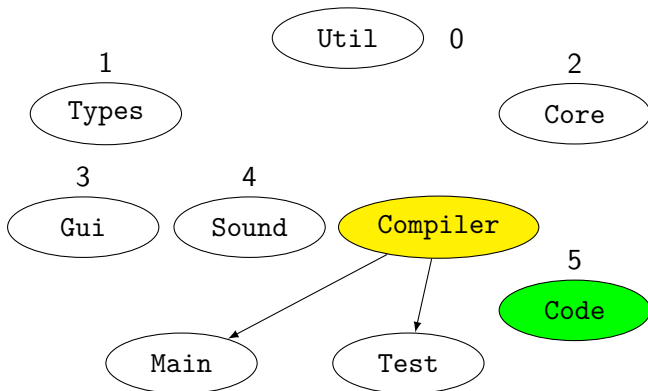
Topologisk sortering – Exempel



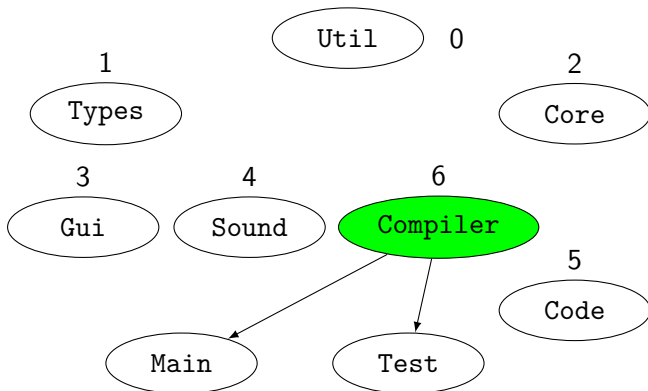
Topologisk sortering – Exempel



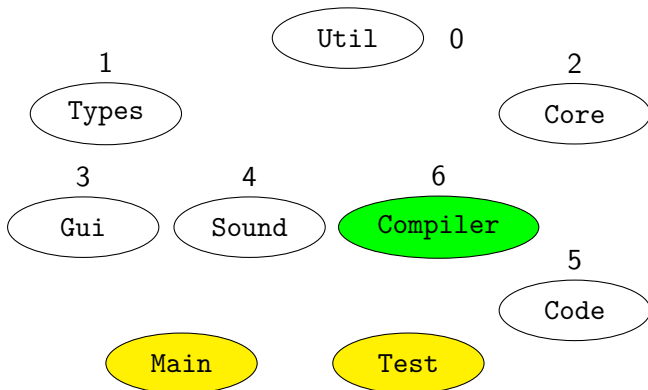
Topologisk sortering – Exempel



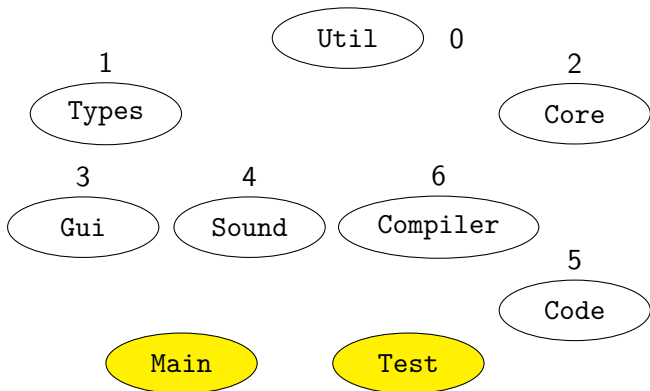
Topologisk sortering – Exempel



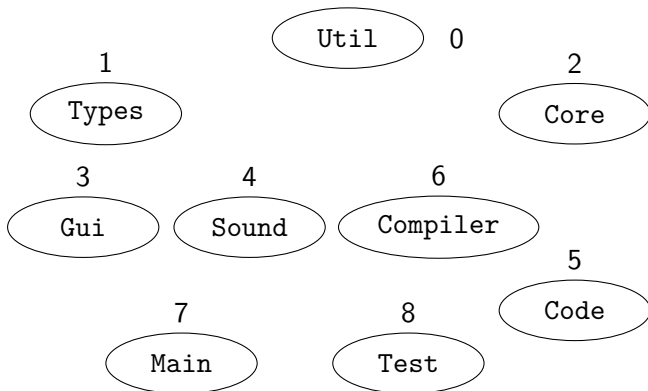
Topologisk sortering – Exempel



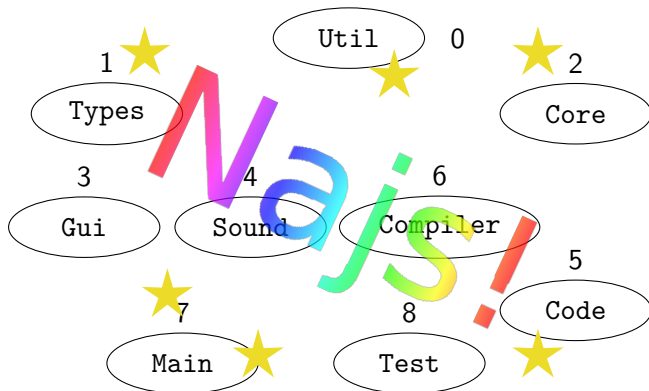
Topologisk sortering – Exempel



Topologisk sortering – Exempel



Topologisk sortering – Exempel

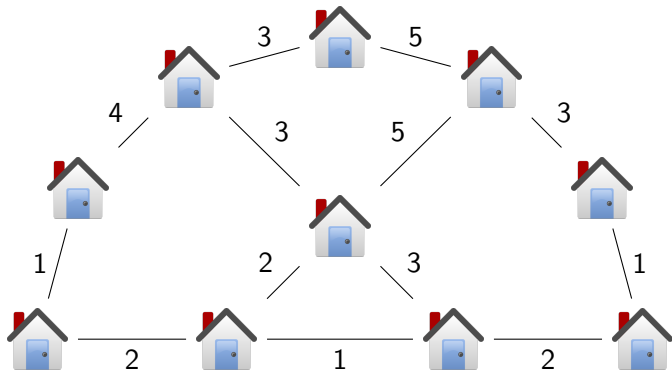


- 1 Topologisk sortering
- 2 **Minsta uppspannande träd**
- 3 Maxflow
- 4 Sammanfattning

Problem

Invånarna i en liten by har bestämt sig för att dra in fiber i alla hus i staden. Du som leder projektet har fått ansvaret för att planera hur fibern ska dras i staden. Du har kommit fram till att det är enklast att dra fibern längs vägar. För att hålla kostnaderna nere, och invånarna glada, vill du gräva upp så lite väg som möjligt. Till din hjälp har du en karta över byn och din trogna dator.

Vilka vägar ska vi använda?



lakttagelser:

Det här är ju en graf!

- Vill välja bågar så att summan blir så låg som möjligt
 - Vill välja tillräckligt många för att koppla ihop alla hus
- ⇒ Vi vill inte ha några **cykler** i grafen
- ⇒ Vi vill göra ett träd, det **minsta uppspännande trädet**

Minsta uppspännande träd

Det finns två välkända algoritmer:

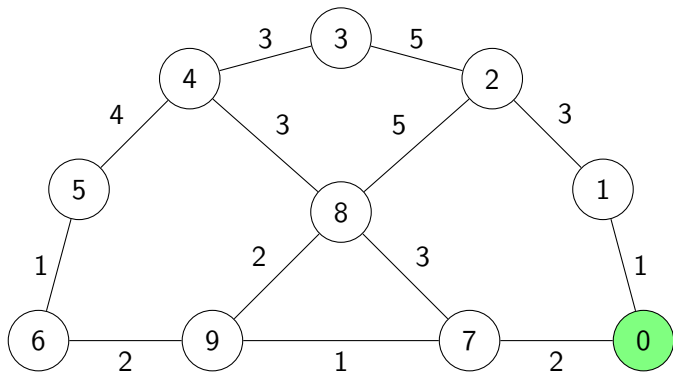
Prims algoritm: $\mathcal{O}(|E| \log(|V|))$

Börja med en nod, utöka sedan trädet successivt genom att lägga till den billigaste bågen som går till en ny nod.

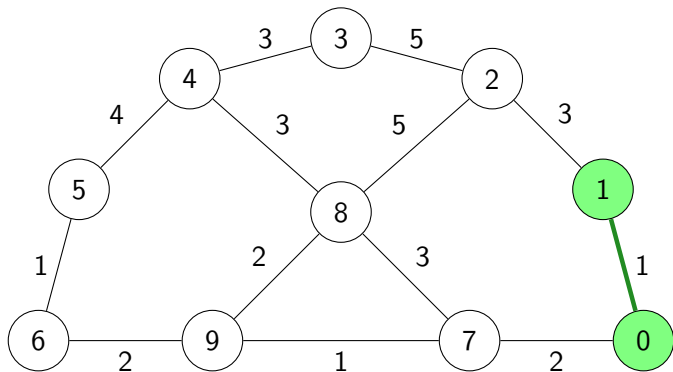
Kruskals algoritm: $\mathcal{O}(|E| \log(|E|)) = \mathcal{O}(|E| \log(|V|))$

Börja med ett tomt träd, utöka sedan trädet med den minsta bågen som inte skapar en cykel till vi fått ett träd.

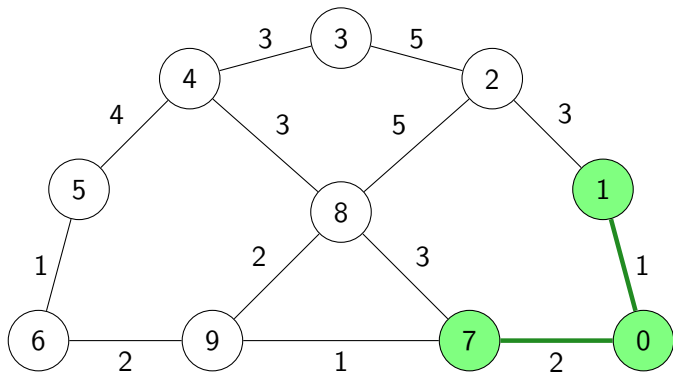
Prims algoritm



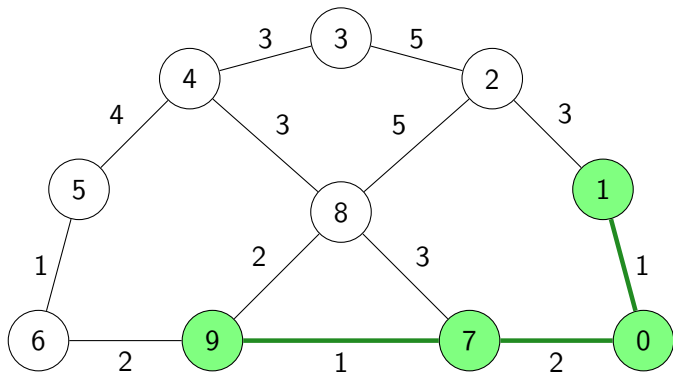
Prims algoritm



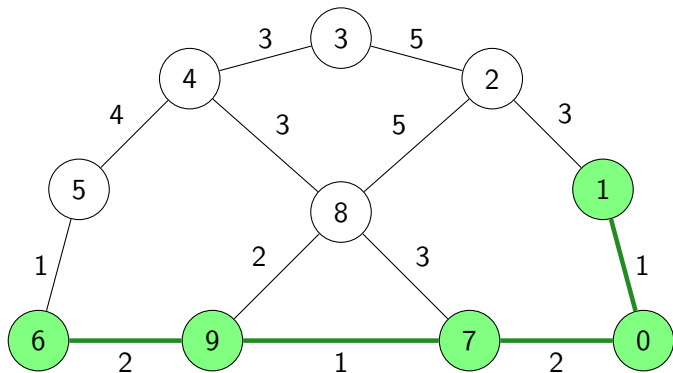
Prims algoritm



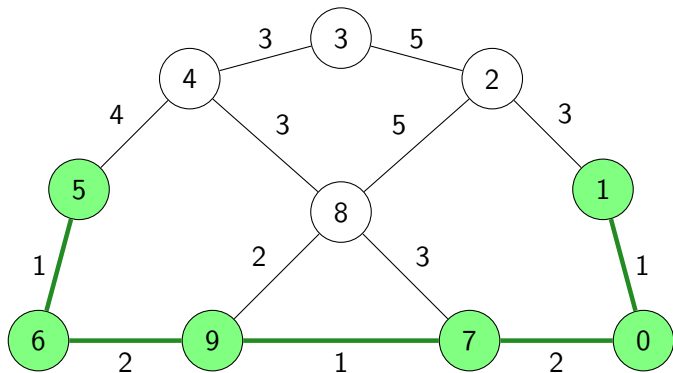
Prims algoritm



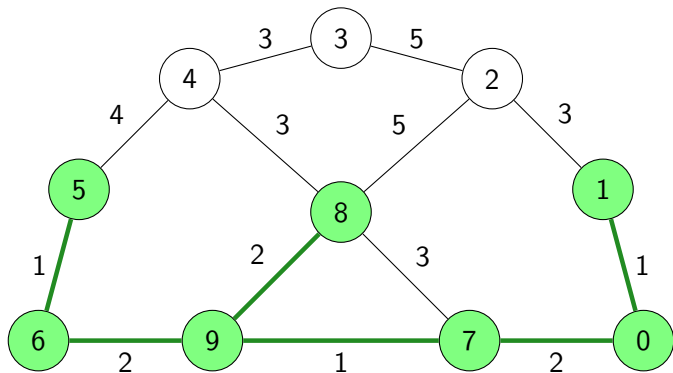
Prims algoritm



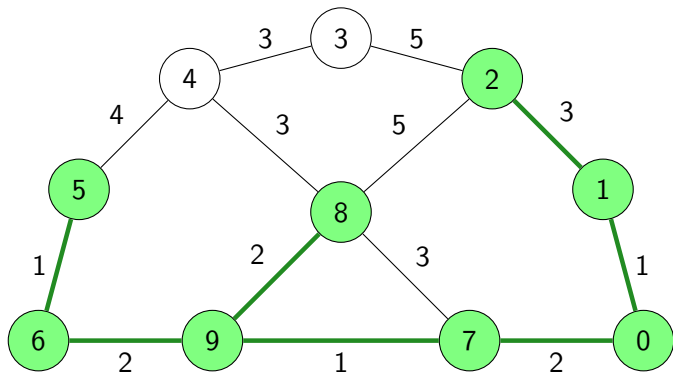
Prims algoritm



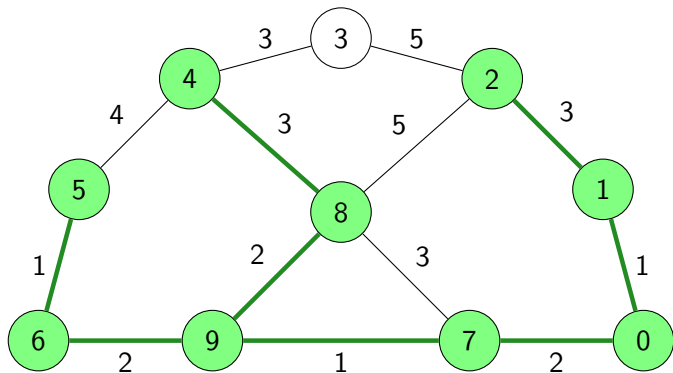
Prims algorithm



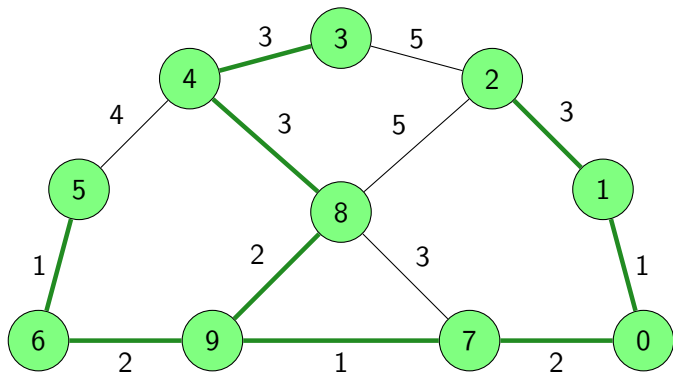
Prims algoritm



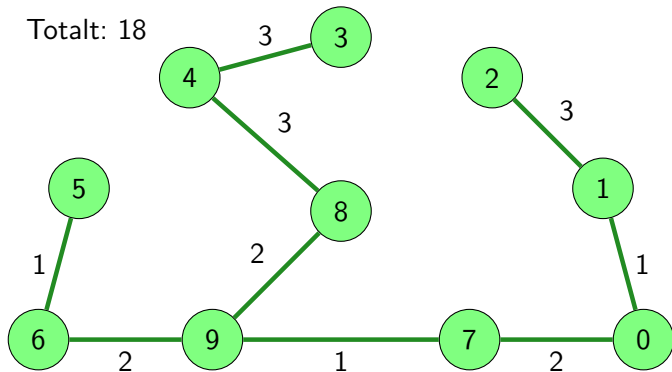
Prims algoritm



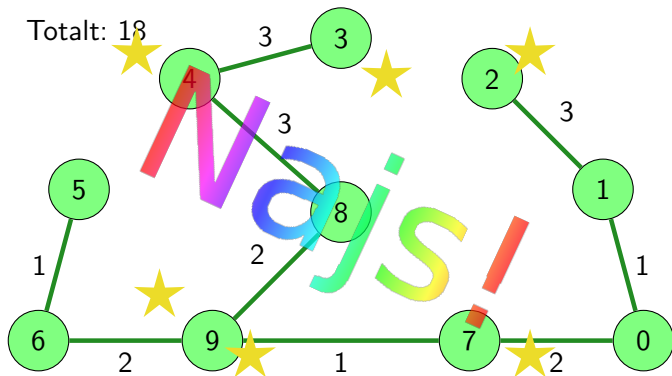
Prims algoritm



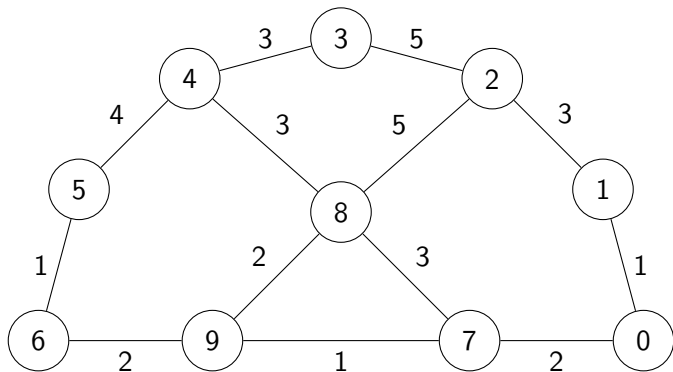
Prims algoritm



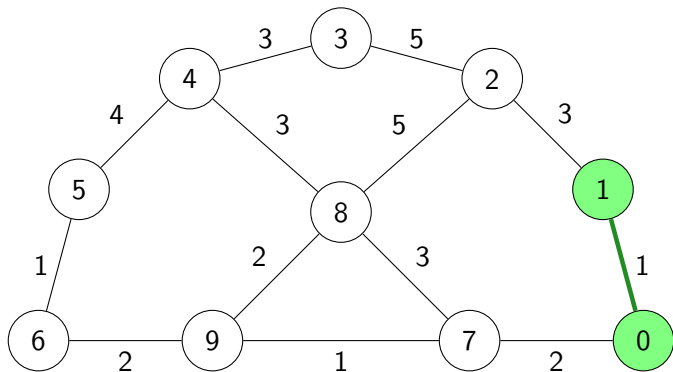
Prims algorithm



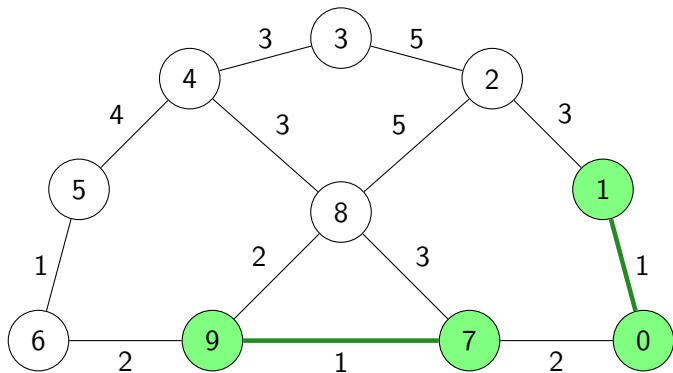
Kruskals algoritm



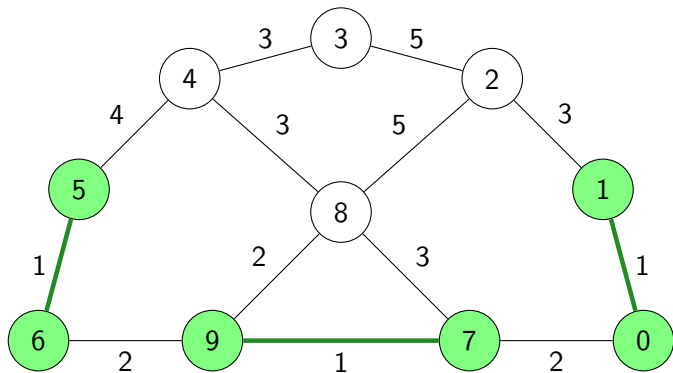
Kruskals algoritm



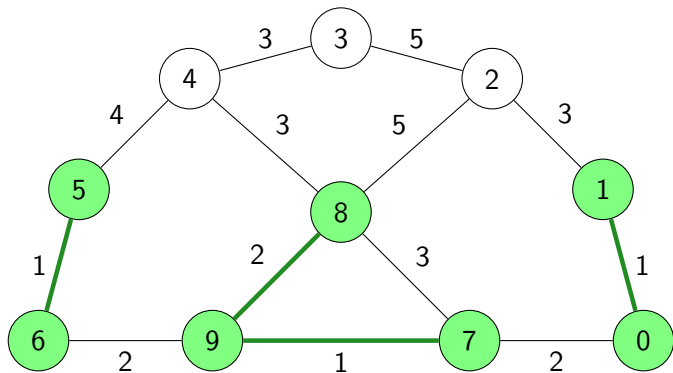
Kruskals algoritm



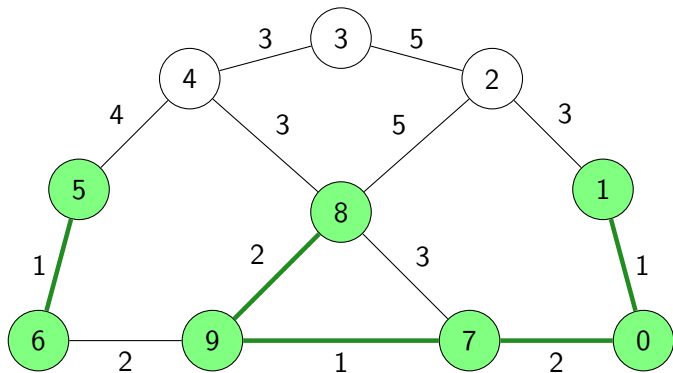
Kruskals algoritm



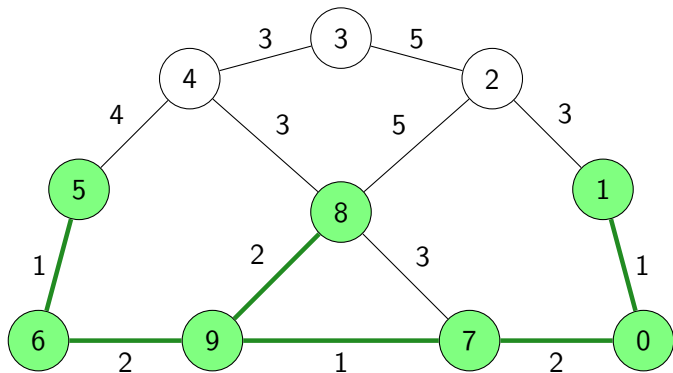
Kruskals algoritm



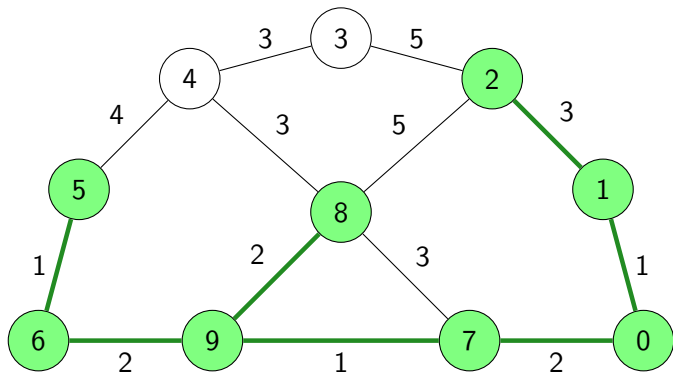
Kruskals algoritm



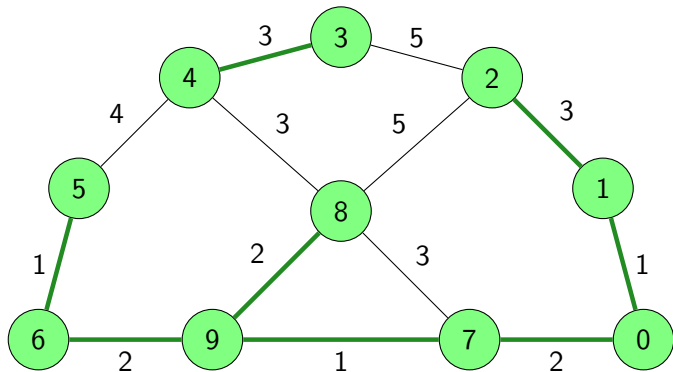
Kruskals algoritm



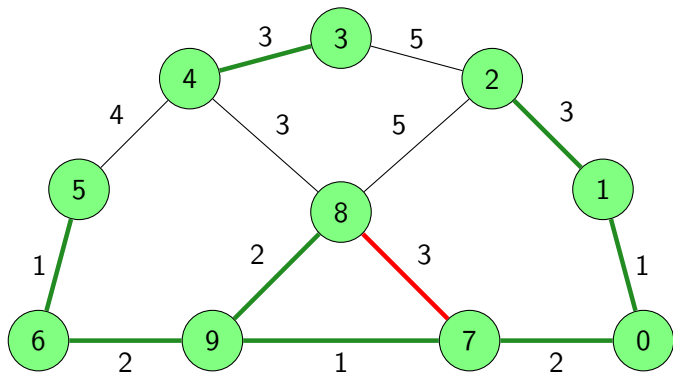
Kruskals algoritm



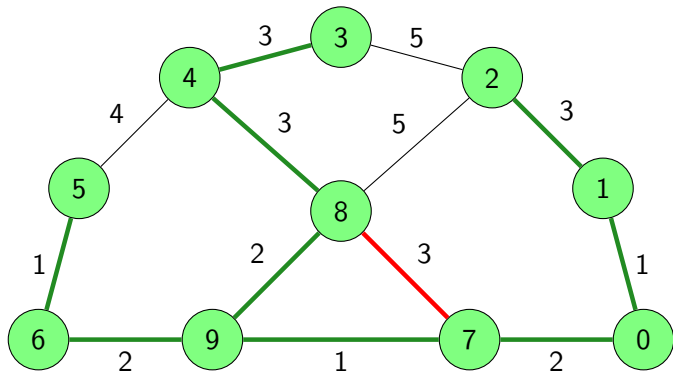
Kruskals algoritm



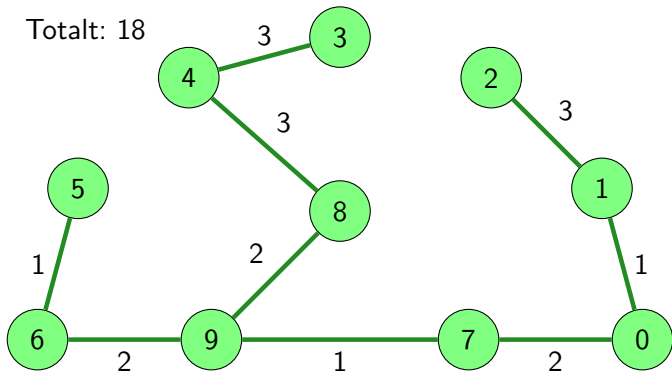
Kruskals algoritm



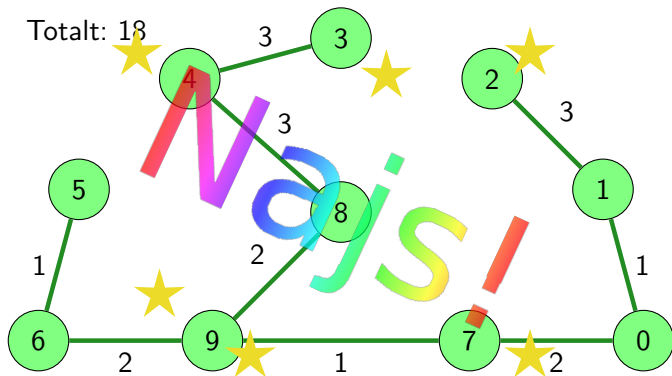
Kruskals algoritm



Kruskals algoritm



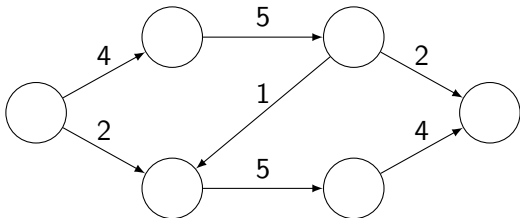
Kruskals algoritm



- 1 Topologisk sortering
- 2 Minsta uppspannande träd
- 3 **Maxflow**
- 4 Sammanfattning

Maxflow – Maximalt flöde i en graf

- Initiera flödet i alla bågar till 0
- Skapa en *residualgraf*
- Hitta en väg (bredden först). Om det finns en väg, updatera flödet och repetera steg 2



Problem

Du ska anordna ett kalas. Du vill ge var och en av gästerna en godispåse, och därför har du köpt lösgodis av diverse sorter som du vet gästerna gillar. Gästerna är olika förtjusta i de olika sorterna, så du vill inte ge för många av en viss typ till en gäst som inte gillar den typen. Du vill dessutom att gästerna ska få lika många bitar totalt.

Hur ska du fördela godiset för att det ska bli så bra som möjligt?

Problem

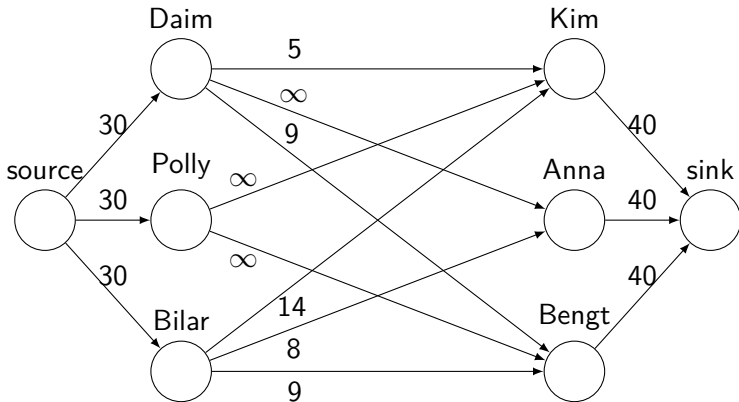
Sort	Person Mängd	Kim 30	Anna 30	Bengt 30
Daim	30	≤ 5		≤ 9
Polly	30		≤ 0	
Bilar	30	≤ 14	≤ 8	≤ 9

Problem

Sort	Person Mängd	Kim 30	Anna 30	Bengt 30
Daim	30	≤ 5		≤ 9
Polly	30		≤ 0	
Bilar	30	≤ 14	≤ 8	≤ 9

Det här är ju en graf!

Grafrepresentation



En lösning

Sort	Person Mängd	Kim 30	Anna 30	Bengt 30
Daim	30	5	22	3
Polly	30	12	0	18
Bilar	30	13	8	9

- 1 Topologisk sortering
- 2 Minsta uppspannande träd
- 3 Maxflow
- 4 **Sammanfattning**

I kursen framöver

- Denna veckan
 - Se till att börja på lab 3
- Nästa vecka
 - Sortering, introduktion till lab 4
- Uppgifter i Kattis
 - brexit (enkel)
Storbritannien lämnar EU, finns det en risk att vårt hemland gör det också?
 - jumpingmonkey (svårare)
Går att lösa med grafsökning, frågan är bara i vilken graf!

Filip Strömbäck

www.liu.se