Ekstra Opgave: Grafer og Algoritmer

Baggrund:

For mere generel forståelse: https://en.wikipedia.org/wiki/Node_(computer_science)

OSPF - Computer Network (3sem bog) kap 5.2 (se Moodle)

A-star Understanding A* Path Algorithms and Implementation with Python | by Adem Akdogan | Towards Data Science

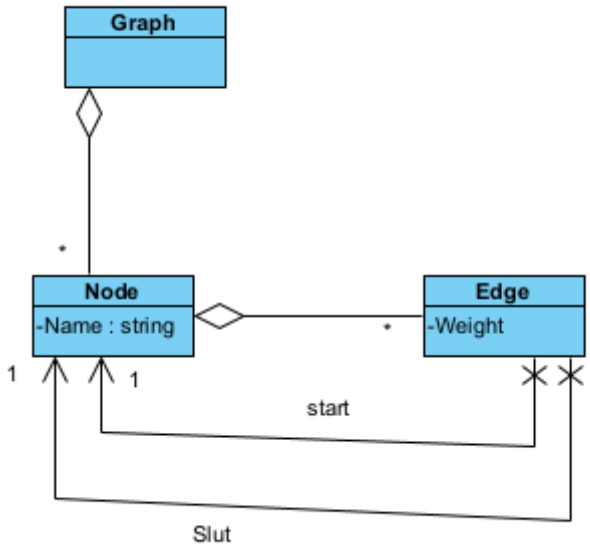
C# implementering af A-star: Astar.cs

Ekstra Del 1: Lav en datastruktur der kan indeholde en graf

Du kan selv definere en struktur, der kan understøtte:

Graf ----> * Node -----> * Link

Evt. følgende DCD:



Ekstra Del 2: Indsæt værdier

Lav en graf – f.eks. med disse værdier:

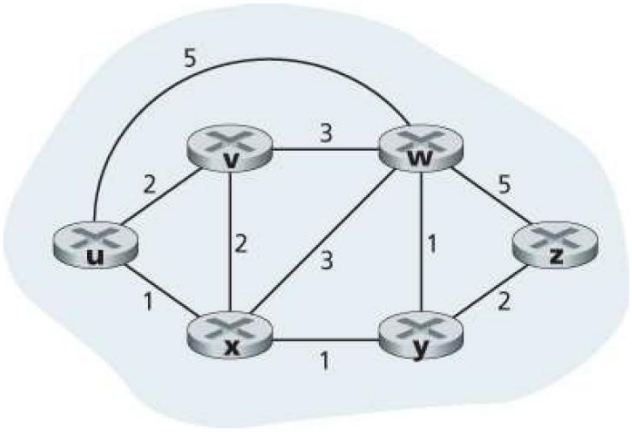


Figure 5.3 Abstract graph model of a computer network

Ekstra Del 3: Lav en klasse OSPFAlg

Lav en klasse OSPFAlg som tager et grafobjekt i konstruktøren.

Lav en metode DoOSPF(Node start), som finder de korteste veje fra start node til alle de øvrige noder (udregner OSPF – eller udspænder et minimalt spanning tree).

Benyt algoritmen fra noten eller fra wikipedia https://en.wikipedia.org/wiki/Dijkstra%27s_algorithm

Ekstra Del 4: Lav en klasse AstarAlg

Det nemmeste er at lave en specialisering af Node klassen fx AStarNode, der indeholder en posistion (x,y)

Lav desuden en klasse Heuristica, der kan estimere afstanden fra en node til en specifik slut-node

Lav en klasse AstarAlg som tager et grafobjekt i konstruktøren.

Lav en metode DoAStar(Node start, Node Slut), som finder de korteste veje fra start node til slut noden.

Benyt evt. wikipedia https://en.wikipedia.org/wiki/A*_search_algorithm eller denne implementering AStar.cs