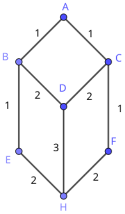
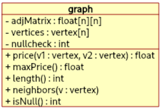
graph



adjacency matrix

	A	B	C	D	E	F	H
A	0	1	1	0	0	0	0
B	1	0	0	2	1	0	0
C	1	0	0	2	0	1	0
D	0	2	2	0	0	0	3
E	0	1	0	0	0	0	2
F	0	0	1	0	0	0	2
H	0	0	0	3	2	2	0

UML



graph

- adjMatrix : float[n][n]
- vertices : vertex[n]
- nullcheck : int

+ price(v1 : vertex, v2 : vertex) : float
+ maxPrice() : float
+ length() : int
+ neighbors(v : vertex)
+ isNull() : int

where

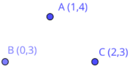
$x,n \in$ int

$x \leq n$

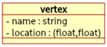
$x \geq 0$

$n \geq 0$

vertices (solmut)



- separate class
- stored separately because:
  - named
  - some function takes as input
  - functions may return sets of vertices (a path)
  - includes location data



vertex

- name : string
- location : (float,float)

edges (kaaret): - in adj matrix, not saved separately

the .neighbors(vertex): vertex[x] function takes a vertex and returns all neighbouring vertices

the .length() function returns n (number of vertices in graph)

the .maxPrice() function returns the biggest value in the adjMatrix

the adjacency martix is sorted according to the order of the vertex array.

the .price() function returns the price of the edge between the two vertices taken from the adjacency matrix.

In the adj matrix: weights between the same vertex = 0, weighs between v1,v2 and v2,v1 identical