Graphs, 14.12

# Термини

## Същината на проблема

* Много често в реалния свят проблемите съдържат обекти, които имат някаква връзка по между си.
* Върхове/възли ребра/връзки
* Не насочен граф, връзките нямат посока (една връзка винаги работи и в 2те посоки) (Приятелства във фейсбук)
* Степен на върха (с колко връзки е свързан върха)
* Насочен граф (връзката между два обекта е само в 1 посока) (Гошо харесва Пенка, но Пенка не харесва Гошо)
* Степен на връх в насочен граф (Incoming степен, Outgoing степен)
* Претеглен граф (върховете имат някаква тежест/мярка)
* Претеглен граф -> Непретеглен (всички връзки са с еднаква тежест)
* Път в граф – последователност от върхове, при което върховете са свързани с връзка
* Прост път – път, в който няма връх, който се повтаря
* Цикличен път в граф – път, който започва от едно място и завършва на същото място
* Цикъл – елемент, който има връзка със себе си
* Multi edge – един връх е свързан с друг с повече от една връзка
* Свързан/Несвързан граф
* Дървото е граф със специални свойства
* Bipartite graph – два отбора, играчите от единия имат връзки с играчите от други и обратното, но нямат връзка по между си.
* Пълен граф – между всяка двойка върхове има връзка
* Tournament граф – пълен граф, но връзките са насочени

# Representing a graph

* Като имаме задача с граф – дават ни броя на върховете - V и броя на ребрата – E
* Ако имаме много върхове – списък на съседи
* Dense graph – много връзки, малко върхове
* Spruce graph – връзки = върхове

## **Edge List**

* Списък със ребра

### Is Edge

* O(E)

### List Edge

* O(E)

### List

* O(E)

## **Adjacency Matrix**

* Матрица N x N

### Is Edge

* O(1)

### List Edge

* O(V2)

### List

* O(V)

## **Adjacency List**

* За всеки връх имаме масив със връзки

### Is Edge – връх X, връх Y

* O(deg(X) + deg(Y))

### List Edge – принтираме всички ребра

* O(E)

### List – връх X

* O(deg(X))

# Exploring Graphs

TOPOLOGICAL SORTING