

ДАА

упр.№10, 4 юни 2020г.

Hello, world!

# Fantastic graphs and where to find them

- ВИДОВЕ

- ориентирани или не

- тегадени или не

- пътни-графи или не (обикновено няма прилики)

- представяне

- матрица на съседство  $\rightarrow$  заема  $\Theta(n^2)$  място

~~добро обхордане на съседите~~

- списъци на съседства  $\rightarrow$  заема  $\Theta(n+m)$  място

foreach  $v$  in  $Adj(u)$

# за DFS и BFS

- имплементации:

- за DFS - по удобни - рекурсивен; може да се направи и без рекурсия (с ръчно управван стек)
- за BFS - на ниво или аналогично на DFS, но с опашка

→ сложност:  $T(n,m) = \Theta(n+m)$        $M(n,m) = \Theta(n)$

- примерни имплементации + още:

<https://github.com/Andreshk/AlgorithmNotes>

# за DFS и BFS

- какво дават като резултат:
  - проверка за достижимост от даден връх
    - може и с двата алгоритъма
  - разстояние (за нетегловни графи) от даден връх до един или вс. достъпни
    - BFS
  - свързаност и свързани компоненти (за неориентирани графи)
    - може и с двата алг - всичко изискване от даден връх обхожда цялата негова съв. компонента
  - двуделност и двуоцветимост
    - бипартитност

# за DFS и BFS

- какво дават като резултат:

- намиране на цикли

- с DFS, гледайки дали някой сосед на текущия връх е съб

- слабо/силно свързани компоненти в ориентирани графи

- слабо (WSC) - за  $\forall i, j \in \text{вещ}$   $\exists$  път от  $i$  до  $j$  или от  $j$  до  $i$

- силно (SCC)  $\rightarrow$  Tarjan за  $T(n, m) = O(n+m)$ ;  $K(n, m) = O(n)$

- срязващи върхове и ребра-мостове (Cut vertices & Bridges)

- за неориентирани

- эксцентрицитет на връх в граф и радиус/диаметър на граф

-  $ecc(u) = \max_{v \in V} d(u, v)$   
 $\downarrow$   
 BFS + лич. обх. на всички с данн.

радиус =  $\min_{u \in V} ecc(u)$   
 диаметър =  $\max_{u \in V} ecc(u)$

1. Пускаме BFS от произв. и търсим най-отдалените  
 2. BFS от този най-отдаленият за дава диаметър



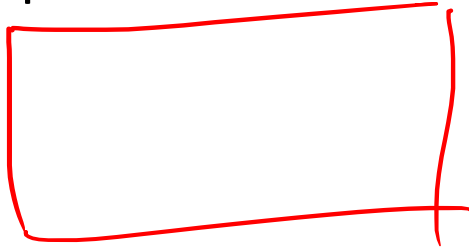
# за DFS и BFS

- какво дават като резултат:

- топологична сортировка на ориентирани графи (+ разпознаване дали такава съществува)

— обратна на реда, в който ~~DFS~~ DFS маршрута върхове с Black

- решаване и генериране на лабиринти



- основа за алгоритми като Ford-Fulkerson, Dijkstra, A\*, Limited DFS, Iterative Deepening

Зад.1. На шахматна дъска се намират кон, топ и царица. С колко най-малко хода може конят да вземе другите фигури?

Зад.2. Задачата за хамстера (за вкъщи)



Зад.3. Да се докаже или опровергае:

- 1 SCC  $\Leftrightarrow$  съществува НР
- 1 SCC  $\Leftrightarrow$  съществува НС
- $>1$  SCC  $\Leftrightarrow$  не съществува НС