Institut for Matematik og Datalogi
Syddansk Universitet, Odense

15. May 2025

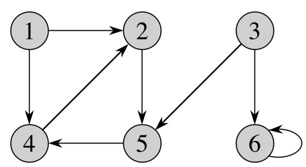
Exercise Sheet Week 13

DSK814

A. Solve in the practice sessions

1. Cormen et al., 4. Edition: Exercise 20.2-1 (page 562) [Cormen et al., 3. Edition: Exercise 22.2-1 (page 601)].

   Show the d and π values that result from $\mathrm{BFS}(G, 3)$ on the directed graph G below, using vertex 3 as the source.

2. Run $\mathrm{BFS}(G, 3)$ on the undirected version of G and provide the resulting d and π values.

3. Cormen et al., 4. Edition: Exercise 20.2-3 (page 562) [Cormen et al., 3. Edition: Exercise 22.2-3 (page 602)].

   Explain why the last line of $\mathrm{BFS}(G, s)$ can be omitted without changing the behavior of the algorithm. This shows that the white and non-white colors are enough, requiring only one bit to store in nodes.

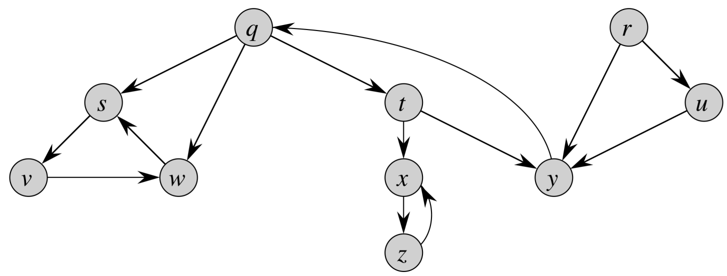
   Hint: Is the gray/black difference used to make decisions in the algorithm?

   Extra task: Explain why this bit is not needed in BFS.

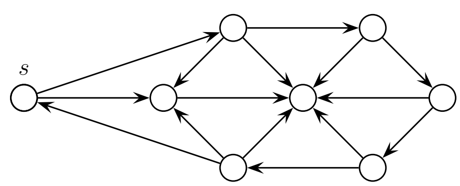
   Hint: look at the d values instead. Why can't the π values be used?

4. Cormen et al., 4. Edition: Exercise 20.3-2 (page 571) [Cormen et al., 3. Edition: Exercise 22.3-2 (page 610)].

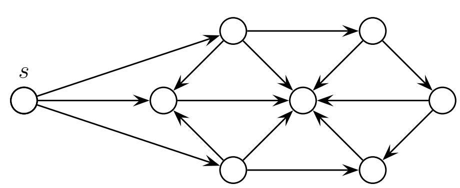
   Run DFS(G, s) on the graph below. Assume that nodes and the neighbor lists are arranged alphabetically in DFS-VISIT(G, u). Specify the resulting d and f values for the nodes and the resulting edge types for edges (tree, back, forward, cross).



5. For all nodes v, provide the distance value $v.d$ assigned by BFS(G, s) in the graph G below.



6. For all nodes v in the graph G below, provide the start time $v.d$ and the end time $v.f$ assigned by DFS(G, s). As the exact result depends on the ordering of the neighbor lists, assume that all node neighbor lists are ordered "clockwise" (starting from "vertically upwards").



7. Cormen et al., 4. Edition: Exercise 20.3-4 (page 571) [Cormen et al., 3. Edition: Exercise 22.3-4 (page 611)].

Explain why the line “u.color = black” in DFS-VISIT(G, u) can be omitted without changing the behavior of the algorithm. This shows that the white and non-white colors are enough, requiring only one bit to store in nodes.

Hint: is the gray/black difference used to make decisions in the algorithm?

Extra task: how can you use the d values in DFS instead of this bit?

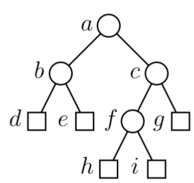
8. Cormen et al., 4. Edition: Exercise 20.3-9 (page 572) [Cormen et al., 3. Edition: Exercise 22.3-10 (page 612)].

   Explain how to extend the pseudocode for DFS to print the type for all edges that are visited in a directed graph G. Then repeat it for an undirected graph G.

   For this extension of DFS, it is necessary for DFS to use all colors white, gray and black, unlike in Exercise 20.3-4 (Exercise 22.3-4).

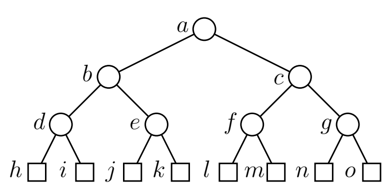
9. Specify a coloring of the nodes that make T_1 a red-black tree.

T_1:



10. List all possible colorings of the nodes that make T_2 a red-black tree.

T_2:



□

11. Provide the solution to the following recurrence.

$$T(n) = 8 \cdot T(n/4) + n^{1.5}$$

12. For each of the following recurrences, indicate if they can be solved using the Master Theorem. If so, state which of the three cases apply and provide the solution.

$$T(n) = 14 \cdot T(n/13) + n$$
$$T(n) = 13 \cdot T(n/13) + n \log n$$
$$T(n) = 14 \cdot T(n/13) + n \log n$$
$$T(n) = 13 \cdot T(n/14) + n$$

B. Solve at home before tutorial in week 14

1. Cormen et al., 4. Edition: Exercise 20.4-3 (page 575) [Cormen et al., 3. Edition: Exercise 22.4-3 (page 615)].

   Give an algorithm whether an undirected graph $G = (V, E)$ contains a simple cycle. Your algorithm should run in time $O(|V|)$, independent of $|E|$.

   Use the fact that if an undirected graph is acyclic, then $|E| \leq |V| - 1$ (this follows from Theorem B.2, items 5 and 6 (page 1170) [Cormen et al., 3. Edition: page 1174]).

2. $(*)$ Cormen et al., 4. Edition: Exercise 20.2-7 (page 563) [Cormen et al., 3. Edition: Exercise 22.2-7 (page 602)].

   An undirected graph $G = (V, E)$ is called *bipartite* if V can be split into two subsets A and B such that all edges have one endpoint in A and the other in B.

   Give an $O(|V| + |E|)$-time algorithm whether G is bipartite. If the answer is yes, also return a possible division into A and B.

   Hint: Extend BFS appropriately.