

IMPORTANT: Write each problem on a separate sheet of paper. Write your name on each sheet. (Why? To speed up the grading we will use parallelization—each problem will be graded by a different TA. At the beginning of the class (on the date when the homework is due) there will be a separate pile for each problem.)

Homework 2 due: Oct. 11 (Thursday); collected before class in Wegmans 1400.

Problem sessions before Homework 2 is due:

Oct. 8 (Monday) 6:15pm - 7:15pm in Hylan 202

Oct. 9 (Tuesday) 6:15pm - 7:15pm in Goergen 109

Oct. 10 (Wednesday) 6:15pm - 7:15pm in Gavett 310

Oct. 10 (Wednesday) 7:40pm - 8:40pm in Hutchinson 473

1 Homework - solve and turn in

2.1 (due Thursday, Oct. 11) Let $G = (V, E)$ be a directed graph (digraph) given in the following adjacency-list representation: for each vertex $v \in V$ we have a linked list of out-neighbors of v . Assume that $V = \{1, \dots, n\}$ and let $m := |E|$.

1. Write pseudocode for a procedure which outputs an adjacency-list representation of the reverse digraph (i. e., G with each edge reversed). The procedure should run in time $O(m + n)$.
2. Write pseudocode for a procedure which outputs the adjacency-list representation of G in which the out-neighbors of each vertex are listed in the increasing order. The procedure should run in time $O(m + n)$.
3. Write pseudocode for a procedure which checks if G is undirected (i. e., the reverse of every $e \in E$ is also in E). The procedure should run in time $O(m + n)$.

2.2 (due Thursday, Oct. 11) Let $G = (V, E)$ be a directed graph; let $n := |V|$, $m := |E|$. Assume that G is not strongly connected. Give an $O(m + n)$ algorithm that decides whether there exist $u, v \in V$ such that adding edge (u, v) to G makes it strongly connected.

2.3 (due Thursday, Oct. 11) We will call a directed graph $G = (V, E)$ **good** if for every pair of vertices $u, v \in V$ there exists a path from u to v or there exists a path from v to u . For example, the following two graphs are good: $(\{1, 2\}, \{(1, 2)\})$ and $(\{1, 2\}, \{(1, 2), (2, 1)\})$. For example, the following graph is not good: $(\{1, 2\}, \{\})$. Give an $O(m + n)$ algorithm that decides whether a given directed graph is good.

2.4 (due Thursday, Oct. 11)

1. We are given a DAG (directed acyclic graph) $G = (V, E)$ and two vertices $u, v \in V$. Assume that $V = \{1, \dots, n\}$ and let $m := |E|$. We want to figure out the length of the longest u - v -path in G (the length of a path v_1, \dots, v_k is $k - 1$). Give an $O(m + n)$ algorithm for the problem. Write pseudocode for the algorithm.
2. We are given a DAG $G = (V, E)$, two vertices $u, v \in V$ and a subset $S \subseteq V$. We want to figure out whether there exists a u - v -path in G that contains a vertex in S (that is, does there exist k and a path v_1, \dots, v_k such that $v_1 = u$, $v_k = v$ and $\{v_1, \dots, v_k\} \cap S \neq \emptyset$). Give an $O(m + n)$ algorithm for the problem. Write pseudocode for the algorithm.

3. We are given a directed graph $G = (V, E)$, two vertices $u, v \in V$ and a positive integer ℓ . We want to figure out whether there exists a u - v -walk of length **at least** ℓ (remember, in a walk we allow repeated vertices and edges; the length of a walk v_1, \dots, v_k is $k - 1$). Give an $O(m + n)$ algorithm for the problem. Clearly describe your algorithm (pseudocode is not required). You can use any algorithm covered in class/textbook (you don't need to write pseudocode for the algorithm used).

2 Additional problems

Try to solve the following problems from Cormen-Leiserson-Rivest-Stein. We will go over the ones that you choose in the problem sessions.

- 22.1-1, 22.1-6, 22.1-7, 22.2-6, 22.2-8, 22.3-9, 22.3-11, 22.5-1, 22.5-3, 22.5-4, 22.5-7,
- 24.1-5, 24.1-6, 24.3-7, 24.3-8, 24.3-9, 24.4-5,
- 25.2-6, 25.2-8,
- 26.2-10, 26.2-11.

Try to solve the following problems from Dasgupta-Papadimitriou-Vazirani. We will go over the ones that you choose in the problem sessions.

- 3.1, 3.2, 3.3, 3.4, 3.6, 3.7, 3.8, 3.9, 3.11, 3.12, 3.13, 3.15, 3.18, 3.19, 3.21, 3.22, 3.23, 3.24, 3.25, 3.26, 3.27,
- 4.1, 4.2, 4.3, 4.4, 4.5, 4.7, 4.8, 4.10, 4.11, 4.13, 4.15, 4.17, 4.18, 4.19, 4.20, 4.21,
- 5.1, 5.2, 5.3, 5.7.

Try to solve the following problems from <http://jeffe.cs.illinois.edu/teaching/algorithms/all-graphs.pdf>. We will go over the ones that you choose in the problem sessions.

- Section 18: problems 9, 10, 11, 12,
- Section 19: problems 1, 5, 6, 7, 9,
- Section 20: problems 1, 2, 3, 4, 5, 6, 7, 8,
- Section 21: problems 4, 5, 6, 7, 9, 11, 12,
- Section 2: problems 3, 4, 5, 6.