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 5 is due: Dec. 7 (Thursday): collected before class in Wegmans 1400.
Problem sessions before Homework 5 is due:
Dec. 4 (Monday) 6:15pm - 7:15pm in Hylan 202
Dec. 5 (Tuesday) 6:15pm - 7:15pm in in Goergen 109
Dec. 6 (Wednesday) 6:15pm - 7:15pm in Gavett 310
Dec. 6 (Wednesday) 7:40pm - 8:40pm in Hutchinson 473

Comprehensive Final Exam is on Dec. 21 (Thursday) 4:00pm - 7:00pm in B&L 109.

1 Homework - solve and turn in

5.1 (due Dec 7) Let \( A \) and \( B \) be two sets. Their sum \( A + B \) is defined to be

\[
A + B := \{a + b | a \in A, b \in B\},
\]

for example if \( A = \{1, 4, 6\} \) and \( B = \{2, 5\} \) then \( A + B = \{3, 6, 8, 9, 11\} \). Give \( O(n \log n) \) algorithm whose input is two sets \( A, B \subseteq \{1, \ldots, n\} \) and the output is set \( A + B \).

5.2 (due Dec 7) We have a weighted directed graph with the following \( n = 3 + 2k \) vertices:

\[
V = \{a, b, c, u_1, \ldots, u_k, v_1, \ldots, v_k\}
\]

and the following \( 4k \) edges:

\[
E = \{(a, u_1), \ldots, (a, u_k), (u_1, b), \ldots, (u_k, b), (b, v_1), \ldots, (b, v_k), (v_1, c), \ldots, (v_k, c)\}.
\]

Each edge-weight is from the set \( \{1, \ldots, n\} \). We want to find out the most common length among the \( a-c \)-paths (if for each \( a-c \)-path we write down its length, what length will occur the most number of times). The input is \( k \) and all the edge weights, that is,

\[
w((a, u_1)), \ldots, w((a, u_k)), w((u_1, b)), \ldots, w((u_k, b)), w((b, v_1)), \ldots, w((b, v_k)), w((v_1, c)), \ldots, w((v_k, c)).
\]

Give a fast algorithm for the problem.

5.3 (due Dec 7) The Hadamard matrices \( H_0, H_1, \ldots \), are defined as follows.

- \( H_0 \) is the \( 1 \times 1 \) matrix \([1]\).
- For \( k > 0 \), \( H_k \) is the \( 2^k \times 2^k \) matrix

\[
H_k = \begin{bmatrix} H_{k-1} & H_{k-1} \\ H_{k-1} & -H_{k-1} \end{bmatrix}.
\]

Show that if \( v \) is a column vector of length \( n = 2^k \) then the matrix-vector product \( H_k v \) can be calculated using \( O(n \log n) \) operations. (Assume that all the numbers involved are small enough that basic arithmetic operations take \( O(1) \) time.)
5.4 (due Dec 7) Consider the following search problem:

INPUT: undirected graph $G = (V, E)$

OUTPUT: labeling $\phi$ of the vertices of $V$ by numbers $\{1, \ldots, |V|\}$ such that

- each vertex receives a different label; and
- for every $\{u, v\} \in E$ the labels of $u$ and $v$ differ by at least 2 (that is, $|\phi(u) - \phi(v)| \geq 2$).

Example #1: for

\[ G = (\{a, b, c, d, e\}, \{\{a, b\}, \{b, c\}, \{c, d\}, \{d, e\}, \{e, a\}\}) \]

$\phi(a) = 1, \phi(b) = 4, \phi(c) = 2, \phi(d) = 5, \phi(e) = 3$ is a valid labeling.

Example #2: for

\[ G = (\{a, b, c, d\}, \{\{a, b\}, \{b, c\}, \{c, d\}, \{d, a\}\}) \]
	no valid labeling exists.

Give a direct\footnote{your reduction may not use any intermediate problem; also you cannot rely on Cook’s theorem.} reduction of the problem to Integer Linear Programming. Clearly describe the variables and constraints in your program.