Worth: 10% **Due:** Tuesday July 2, 11:59pm.

Remember to write your *full name*, and *student number*.
Please make sure that the work you submit does not contain someone else's words or ideas. Then, to protect yourself, list on the front of your submission ***every*** *source of information you used to complete this homework (other than your own lecture and tutorial notes, and materials available directly on the course webpage). For example, indicate clearly the* ***name*** *of every student with whom you had discussions, the* ***title*** *of every additional textbook you consulted, the* ***source*** *of every additional web document you used,* etc.

For each question, please write up detailed answers carefully. Make sure that you use notation and terminology correctly, and that you explain and justify what you are doing. Marks **will** be deducted for incorrect or ambiguous use of notation and terminology, and for making incorrect, unjustified, ambiguous, or vague claims in your solutions. In any of your answers you can use any algorithm we discussed in class without proving it solves the problem we discussed in class optimally. If we discussed the runtime of the algorithm you can also use that without reproving it. The same goes for any Lemma, Theorem or Fact we discussed in class.
Note that any algorithm you propose should have a polynomial run time (algorithms with exponential run time such as brute force do not earn any point).
You will receive 20% in each question if you leave it completely blank or say "I don't know".

---

[15] 1. Consider the following variant of the knapsack problem. The input consists of n items $\{I_1, ..., I_n\}$ and a knapsack size bound C. As in the standard knapsack problem, an input item I_j is a pair (w_j, v_j). Now a feasible subset of items is one satisfying the knapsack bound and such that each item either does not occur or occurs an odd number of times. That is, a feasible set of item indices is a vector $(k_1, ..., k_n)$ such that $\sum_j k_j w_j \leqslant C$ and $k_j \in \{0, 1, 3, 5, ...\}$. The goal is to maximize $\sum_j k_j v_j$ for a feasible set. Assume further that $C = n^2$ and that $n \leqslant w_j \leqslant n^2$ for all j. Provide a dynamic programming algorithm for this problem and analyze its asymptotic complexity as a function of n.

[15] 2. Exercise 6.6 of the DPV textbook.

[10] 3. Exercise 6.19 of the DPV textbook.

[15] 4. Exercise 6.20 of the DPV textbook.

[10] 5. Exercise 7.10 of the DPV textbook.

[15] 6. In a flow network, an edge is called a bottleneck edge if increasing its capacity results in an increase in the maximum flow. Give an efficient algorithm to identify all bottleneck edges in a network.

[10] 7. After an earthquake, n injured people have been identified across a region. These people need to be rushed to the k hospitals existing in this region. Each person should be brought to a hospital that is within a 1 km distance. Each hospital can receive at most $\lceil n/k \rceil$ people. Given the location of these injured people and the hospitals, propose a polynomial time algorithm that determines whether this is possible.