Due: Wednesday, November 15, 2017, 10AM on MarkUs

You will receive 20% of the points for any (sub)problem for which you write "I do not know how to answer this question." You will receive 10% if you leave a question blank. If instead you submit irrelevant or erroneous answers you will receive 0 points. You may receive partial credit for the work that is clearly "on the right track." .

1. (20 points)

   Suppose we have a maximum integral flow f with $val(f) \geq 1$ in a flow network $\mathcal{F} = (G, s, t, c)$ with integral capacities.

   - (5 points) Does there always exist an edge e such that by decreasing the capacity of e by one unit to $c(e) - 1$, the value of the maximum flow is decreased by exactly one unit? Justify your answer.
   - (5 points) Does your answer to the above question depend on the assumption that all capacities are integral?
   - (10 points) Suppose that there is an edge e such that increasing its capacity by one unit will result in the increase of the maximum flow value. Show how to find such an edge in time that is determined by the computation of a least cost paths algorithm.

2. (20 points) Consider the following makespan problem in the restricted machines model. The input is a set $\mathcal{J}$ of unit processing time jobs $\{J_1, \ldots, J_n\}$ where each job J_j can be scheduled on some subset $S_j \subseteq \mathcal{M} = \{M_1, \ldots, M_m\}$ of the m identical machines. A schedule is a mapping $\sigma : \mathcal{J} \rightarrow \mathcal{M}$ such that $\sigma(J_j) = M_i$ implies $M_i \in S_j$. The objective is to compute a schedule σ so as to minimize the *makespan value* which is $max_i|j : \sigma(J_j) = M_i|$. That is, to minimize the latest completion time (= number of jobs) for all machines.

   - (10 points) Show that the online greedy algorithm is not optimal. That is, there is a sequence of jobs on which the online greedy algorithm does not produce an optimal schedule. The online greedy algorithm schedules jobs on the least loaded machine and breaks ties in favour of the machine with the smallest index. For a small bonus, can you show that for arbitarily large m, there is a set of m jobs that can be optimally scheduled to have makespan value = 1, while the greedy algorithm will schedule $\log_2 m$ jobs on one machine?
   - (10 points) Show how to optimally solve this makespan problem by reducing the problem to optimal flows. Hint: First use flows to determine the optimal makespan value. Justify the optimality of your algorithm.

3. (15 points) Consider the *graph 3-colourability* problem.

   - (10 points) Show how to reduce the search problem to the decision problem. That is, how to find a valid 3-colouring (when one exists) given that there is a subroutine for determining if a graph has a 3-colouring. Hint: you should try to create the colouring one node at a time using one of the ideas in the proof of the transformation of 3SAT to 3-colouring that established the NP completeness of the 3-colouring decision problem.

- (5 points) Assume that your subroutine for the decision problem takes time $T(n, m)$ where $n = |V|, m = |E|$. What is the asymptotic time required for the search algorithm?

4. (10 points) Consider the following Set Selection decision problem:
Input: Given an integer $k \geq 1$ and a collection of sets $\{C_1, \ldots, C_n\}$ where each $C_i \subseteq \{1, 2, \ldots, m\}$.
Decision problem: Does there exist a subset of indices $S \subseteq \{1, 2, \ldots, n\}$ such that $|S| = k$ and $C_i \cap C_j = \emptyset$ for all $i, j \in S, i \neq j$?

   Show that Set Selection is NP complete by showing that it is in NP and that it is NP hard by giving a polynomial time transformation from Independent Set.

5. (20 points) Consider the following Unit Value-One Machine scheduling decision problem:
Input: Given an integer $k \geq 1$ and a collection of *unit value* jobs $\{J_1, \ldots, J_m\}$ where each job J_i is described by a triple of integers (r_i, p_i, d_i) where $r_i \geq 0$ is the release time, p_i is the processing time, and d_i is the deadline for the job. That is, if J_i is scheduled then it must be scheduled to start at some time $t_i : r_i \leq t_i \leq d_i - p_i$.
Decision problem: Does there exist a subset S of k jobs that can be scheduled within their deadlines on one machine so that no two jobs in S are overlapping when scheduled?

   - (10 points) Assuming $r_i = 0$ for all i, show that this problem can be solved by a dynamic programming algorithm. algorithm. Hint: The optimization problem (i.e., scheduling the maximum number of jobs possible) can be solved optimally by a greedy algorithm.
   - (10 points) Show that when the release times are part of the problem (i.e. not assumming that all $r_i = 0$), that the Unit Value-One Machine scheduling problem is NP complete. That is, show that it is in NP and show that it is NP hard by providing a polynomial time transformation from one of the following decision problems L shown to be NP complete in class: 3SAT, Independent Set, 3 Colour, and Subset Sum.
   Hint: Think of how you might use a job of unit length to partition the jobs.