You may work in a group of at most 3 students. Carefully read [http://engr.course.illinois.edu/cs473/policies.html](http://engr.course.illinois.edu/cs473/policies.html) and [http://engr.course.illinois.edu/cs473/integrity.html](http://engr.course.illinois.edu/cs473/integrity.html).

One member of each group should submit via Gradescope.

1. [13 pts]
   
   (a) [10 pts] Run the simplex method on the following linear program:

   \[
   \begin{align*}
   \text{maximize} & \quad x_1 + x_2 \\
   \text{s.t.} & \quad 3x_1 + 2x_2 \leq 27 \\
   & \quad x_2 \leq 8 \\
   & \quad x_1 + 5x_2 \leq 35 \\
   & \quad 2x_1 + x_2 \leq 17 \\
   & \quad x_1, x_2 \geq 0.
   \end{align*}
   \]

   Start with the initial basic solution \((x_1, x_2) = (0, 0)\), and choose \(x_1\) as the entering variable in the first iteration. Show the new slack form after every iteration.

   (b) [3 pts] Write down the dual of the linear program from (a). What is the optimal dual solution?

2. [20 pts] For two points \(p = (p_1, \ldots, p_d) \in \mathbb{R}^d\) and \(q = (q_1, \ldots, q_d) \in \mathbb{R}^d\), their rectilinear distance is defined as

   \[ D(p, q) = |p_1 - q_1| + \cdots + |p_d - q_d|. \]

   In the rectilinear 1-center problem in \(d\) dimensions, we are given a set \(P\) of \(n\) points in \(\mathbb{R}^d\), we want to find a point \(q \in \mathbb{R}^d\) (not necessarily in \(P\)) that minimizes \(\max_{p \in P} D(p, q)\).

   (a) [8 pts] First show that the problem can be solved in \(O(n)\) time for any constant dimension \(d\). How does the running time of your algorithm grow as a function of \(d\)?

   (Hint: find a small number of candidate points in \(P\) that could be the farthest point from any \(q\)…)

   (b) [12 pts] Show how to solve this problem for large (nonconstant) dimensions \(d\) by using linear programming. The number of variables and constraints should be polynomial in \(n\) and \(d\). (Remember to justify correctness of your reduction.)

3. [12 pts] We are given \(n\) tasks to perform. Task \(i\) requires \(p_i\) units of power consumption for a duration of \(h_i\) hours. At any moment in time, we can perform at most 3 different tasks, and at any moment in time, the total power consumption must be at most \(P\). A task may be preempted (possibly multiple times) at no extra cost. The problem is to devise a schedule to perform all \(n\) tasks with the minimum total number of hours.

   For example: for \(n = 5\) with \(p_1 = 10, h_1 = 8.5, p_2 = 20, h_2 = 9, p_3 = 60, h_3 = 4, p_4 = 80, h_4 = 3.5, p_5 = 90, h_5 = 2,\) and \(P = 100\), one feasible solution is to do tasks 1 and 5 for 2
hours, then tasks 2 and 4 for 3.5 hours, then tasks 1, 2, and 3 for 4 hours, then tasks 1 and 2 for 1.5 hours, and finally task 1 for 1 hour; the total number of hours is 12. (I did not check if this is optimal. Also, for this small example, the constraint that we can do at most 3 tasks at any time is not important; but it could make a difference on larger instances.)

Describe how to solve this problem using linear programming. The number of variables and constraints should be polynomial in $n$. 