Solve all questions by hand and show your work.

1. (2 points) Consider the following (directed) graph \( G = (V, E) \) with capacities \( c_e \) given on each edge. Let \( s = 1 \) and \( t = 6 \).

   \[ \begin{array}{c}
   1 & 2 & 3 & 4 & 5 & 6 \\
   3 & 6 & 10 & 5 & 3 & 2 \\
   1 & 3 & 3 & 10 & 5 & 6 \\
   \end{array} \]

   (a) Find a maximum \( s \)-\( t \) flow in \( G \). What is the value of this flow?
   (b) Find a minimum (directed) \( s \)-\( t \) cut in \( G \). What is the value of this cut?

2. (2 points) Consider the following problem faced by a hospital that is trying to evaluate whether its blood supply is sufficient.

   The basic rule for blood transfusion is the following: A person’s blood has certain antigens, and a person cannot receive blood with a particular antigen if their own blood does not have this antigen. Concretely, this principle underpins the division of blood into four types: A, B, AB, and O. Blood of type A has the A antigen, blood of type B has the B antigen, blood of type AB has both, and blood of type O has neither. Thus, patients with type A can receive only blood types A or O, patients with type B can receive only B or O, patients with type O can receive only O, and patients with type AB can receive any of the four types.

   **Input:** The input consists of 9 integers: an integer \( k > 0 \), the hospital’s supply in whole units of the different blood types \( 0 \leq s_O, s_A, s_B, s_{AB} \leq k \), and the projected demand of each blood type for the coming week \( 0 \leq d_O, d_A, d_B, d_{AB} \leq k \).

   **Goal:** Design an \( O(k) \) algorithm to evaluate if the blood on hand would suffice for the projected need. Prove the correctness of your algorithm and analyze its runtime.

3. (2 points) Decide whether the following statements are true or false. Justify your answer.

   (a) If \( X \leq_P Y \) and \( Y \in \mathbf{P} \), then \( X \in \mathbf{P} \).
   (b) If \( X \leq_P Y \) and \( X \) is \( \mathbf{NP} \)-complete, then \( Y \) is \( \mathbf{NP} \)-complete.
4. (extra credit, 1 point) (This question was written by Prof. David Kempe and appeared in the Fall 2012 Programming Contest at the University of Southern California. The theme of the Fall 2012 Contest was “The Mars Rover Contest”.)

It is well known that Martians are huge pranksters. Once they figured out that this weird new vehicle on their planet was supposed to take pictures for scientific analysis, they decided that it would be a great sport to try and confuse the Earth scientists as much as possible. The idea is that they will congregate in large numbers just outside the image taken by Curiosity. Between two successive images, they will quickly race in, move some of the rocks in the picture around, and then disappear again. So when Curiosity takes two pictures of the same spot 30 seconds apart, scientists see two different images. Scientists on Earth of course know of these tendencies of Martians. From pairs of pictures, they are trying to learn now just how fast Martians can run, even though they’ve never seen one.

**Input:** There are two pictures, taken \( t \geq 1.0 \) seconds apart. The first picture is described by the locations of \( n \geq 1 \) rocks as pairs of coordinates \((a_i, b_i)_{i=1}^{n}\). The second picture is similarly described by the location of \( n \) rocks \((x_j, y_j)_{j=1}^{n}\). The two images will always have the same number of rocks. Note that all the rocks look the same so they are indistinguishable in the images. All coordinates are real numbers between 0 and 1, inclusive.

There is a really large number of Martians gathered just around the unit square \([0, 1] \times [0, 1]\). They can each run at the same speed \( v \), which is measured in units per second. Right after the first picture is taken, for each rock that the Martians decide to move, a Martian closest to the rock will run straight to that rock. Then, he will move it to its new location, for which his speed is only \( v/2 \) (because he is pushing a heavy weight). From the new location, he will run out of the picture at a speed of \( v \), taking the shortest route out, before the next picture is taken. If a rock is not to be moved between the two pictures, then of course no Martian needs to run to it.

**Goal:** Design an \( O(n^3) \) algorithm to infer from the two pictures how fast the Martians are running. More precisely, you are to find the smallest speed \( v \) such that it is possible for the Martians collectively to follow the protocol above and move all the rocks without being detected. You only need to output \( v \) rounded up to two decimals. (For example, if \( v = 0.38 \) is fast enough but \( v = 0.37 \) is too slow, then your algorithm should output \( v = 0.38 \).)

(Hint: binary search for the smallest possible \( v \). Given a speed \( v \), how can we decide which rock in the first image corresponds to which rock in the second image?)