CECS 329, Solutions to LO5 Assessment, 10-13, Fall 2022, Dr. Ebert

NO NOTES, BOOKS, ELECTRONIC DEVICES, OR INTERPERSONAL COMMUNICATION ALLOWED. Submit each LO solution on ONE SIDE of a single sheet of paper.

Problems

LO1. An instance of the `Simultaneous Incongruences` decision problem is a set

$$S = \{(a_1, b_1), (a_2, b_2), \ldots, (a_n, b_n)\}$$

of pairs of positive integers, where $a_i \leq b_i$ for every $i = 1, \ldots, n$. The problem is to decide if there is a positive integer x for which

$$x \not\equiv a_i \bmod b_i,$$

for every $i = 1, \ldots, n$.

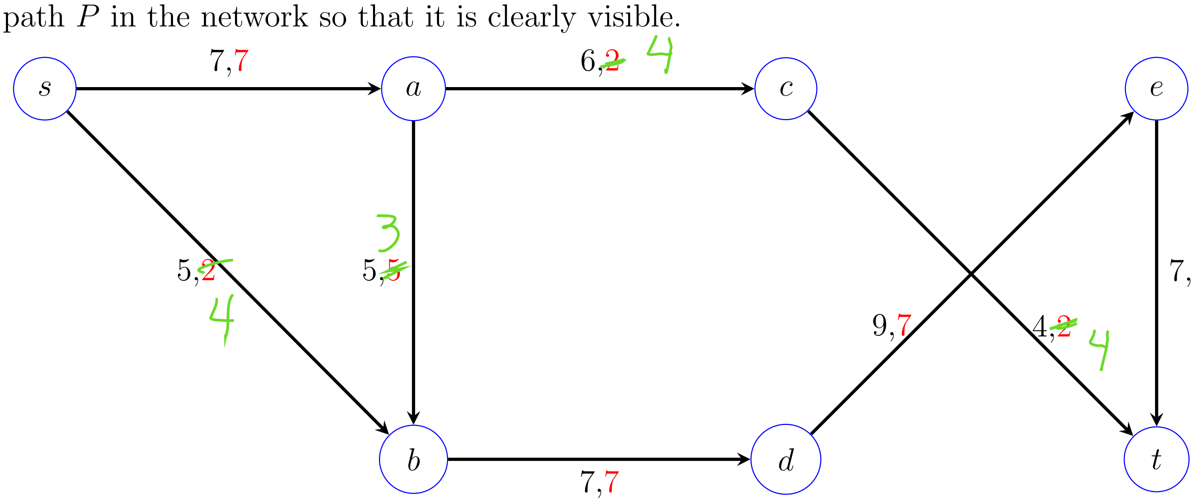
(a) Verify that $S = \{(1, 2), (2, 3), (3, 5), (4, 7), (5, 11), (6, 13)\}$ is a positive instance of `Simultaneous Incongruences`.
Solution. $x = 10$ satisfies $x \not\equiv a_i \mod b_i$, for each $i = 1, \ldots, 6$.

(b) Provide two size parameters for the `Simultaneous Incongruences` problem. Hint: you may assume b_n is the largest integer occurring in any pair in S.
Solution. n: number of pairs. $\log b_n$: bound on the number of bits required by each integer.

(c) Rocky claims that he has discovered an algorithm for deciding `Simultaneous Incongruences` whose running time is worst-case quadratic and includes both size parameters. Provide a big-O expression that accurately represents Rocky's claim.
Solution. $O(n \log b_n)$.

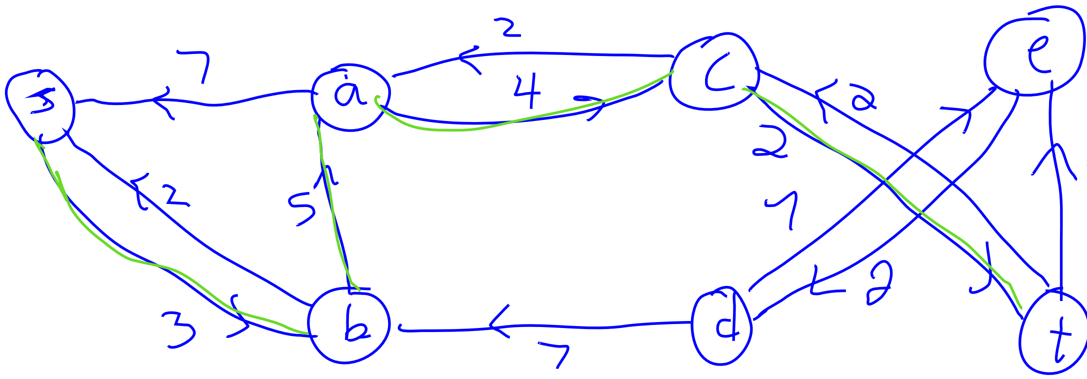
LO2. A flow f (2nd value listed on each edge) has been placed in the network G below.

(a) Draw the residual network G_f and use it to determine an augmenting path P. Highlight path P in the network so that it is clearly visible.



Answer.

Augmenting Path: s, b, a, c, t
$x = \min(3,5,4,2) = 2$



(b) In the original network, cross out any flow value that changed, and replace it with its updated value from $f_2 = \Delta(f, P)$.

Answer. See the original network graph.

(c) What one query can be made to a `Reachability` oracle to determine if f_2 is a maximum flow for G? Hint: three inputs are needed for the `reachable` query function. Cleary define each of them.

Answer. `reachable`(G_{f_2}, s, t).

LO3. Answer the following questions.

(a) Provide the definition of what it means to be a mapping reduction from decision problem A to decision problem B.

Answer. See page 1 of Mapping Reducibility lecture.

(b) As part of her network security project, Laura is working with a simple graph H that has 78 vertices and 232 edges. She needs to know whether or not H has a Hamilton path. Her colleague Simon has implemented the Python function

```
Boolean has_LPath(Graph G, int k);
```

that decides if the input graph G has a simple path of length k. Explain how Laura can use Simon's function to determine whether or not H has a Hamilton path.

Answer. Laura can call `has_LPath` with inputs H and $k = 77$, since a graph will have a Hamilton path iff it has a simple path of length $|V| - 1$.

LO4. Recall that an instance of the `Hamilton Cycle (HC)` decision problem is a simple graph $G = (V, E)$ and the problem is to decide if G has a Hamilton cycle, i.e. a path P that visits every vertex of G exactly once before returning to P's start vertex. We now establish that `HC` is a member of `NP`.

(a) Define a certificate for verifying that an instance of `HC` is positive. Hint: recall that a path can be represented as a sequence of vertices.

Answer. Certificate P is a sequence of $n = |V|$ vertices $v_1, v_2, \ldots, v_{n-1}, v_n$.

(b) Provide a semi-formal verifier algorithm for `HC`.

Answer.

Create an *empty* lookup table T.
For each $i = 1, \ldots, n - 1$,
  If $(v_i, v_{i+1}) \notin E$, then return 0.
  If $v_i \in T$, then return 0. $//v_i$ occurs more than once in P
  Insert v_i into T.
If $v_n \in T$, then return 0. $//v_n$ occurs more than once in P
If $(v_n, v_1) \notin E$, then return 0.
Return 1.

(c) Provide size parameters for `HC` and clearly define each one. Hint: there are two of them.

Answer. $m = |E|$, $n = |V|$.

(d) Provide the verifier's running time. Justify your answer.

Answer. After creating an edge lookup table in $O(m)$ steps, $O(n)$ steps are required to verify that each (v_i, v_{i+1}) is an edge of G. Also, it takes $O(n)$ steps to i) check that each vertex v does not appear in T, followed by inserting v into T. This gives a total running time of $O(m+n)$.

LO5. An instance $\mathcal{C}$ of `3SAT` consists of clauses $c_1 = (x_1, \overline{x}_2, x_3)$, $c_2 = (\overline{x}_2, x_3, x_4)$, $c_3 = (\overline{x}_1, x_2, \overline{x}_4)$, and $c_4 = (\overline{x}_1, \overline{x}_3, x_4)$. Answer the following questions about the mapping reduction $f(\mathcal{C}) = (G, k)$ provided in lecture from `3SAT` to `Clique` and applied to instance $\mathcal{C}$.

(a) How many vertices and edges does G have? Explain and show work. Hint: there are six different vertex-group pairs.

Answer. $|V| = (3)(4) = 12$, while $|E| = 9 + 7 + 7 + 7 + 8 + 8 = 46$ is the number of consistent pairs of vertices that come from different groups.

(b) What is the value of k?

Answer. $k = 4$, the number of clauses.

(c) Given that $\alpha = (x_1 = x_2 = 0, x_3 = 1, x_4 = 0)$ satisfies $\mathcal{C}$, provide a clique set for G that certifies (G, k) is a positive instance of `Clique`. Hint: for each clique member, indicate the group from which it came.

Answer. $C = \{\overline{x}_2, \overline{x}_2, \overline{x}_1, \overline{x}_1\}$ is a 4-clique, where the i th literal in C comes from group i, $i = 1, 2, 3, 4$. Since C is a consistent set of literals, and each literal comes from a different group (i.e. clause), it follows that C forms a clique in G.