## COMP 3804 — Solutions Tutorial February 9

**Problem 1:** You are given three beer barrels  $B_1$ ,  $B_2$ , and  $B_3$ . Barrel  $B_1$  has a capacity of 8 litres, barrel  $B_2$  has a capacity of 5 litres, and barrel  $B_3$  has a capacity of 3 litres.

At any moment, each barrel contains a given amount of beer (in litres). In one *step*, you can pour beer from one barrel, say  $B_i$ , to another barrel, say  $B_j$ . This step terminates at the moment when  $B_i$  becomes empty or  $B_j$  becomes full, whichever happens first.

To give some examples:

- If  $B_1$  contains 6 litres of beer,  $B_2$  contains 2 litres of beer, and  $B_3$  contains 0 litres of beer, then we can pour the entire contents of barrel  $B_2$  to barrel  $B_3$ . At the end of this step,  $B_1$  contains 6 litres of beer,  $B_2$  contains 0 litres of beer, and  $B_3$  contains 2 litres of beer.
- If  $B_1$  contains 3 litres of beer,  $B_2$  contains 4 litres of beer, and  $B_3$  contains 1 litre of beer, then we can pour 2 litres of beer from  $B_1$  to  $B_3$ . At the end of this step,  $B_1$  contains 1 litre of beer,  $B_2$  contains 4 litres of beer, and  $B_3$  contains 3 litres of beer.

## **Decision problem:**

- Let  $b_1$ ,  $b_2$ , and  $b_3$  be integers such that  $b_1 \ge 0$ ,  $b_2 \ge 0$ ,  $0 \le b_3 \le 3$ , and  $b_1 + b_2 + b_3 = 4$ . Similarly, let  $b'_1$ ,  $b'_2$ , and  $b'_3$  be integers such that  $b'_1 \ge 0$ ,  $b'_2 \ge 0$ ,  $0 \le b'_3 \le 3$ , and  $b'_1 + b'_2 + b'_3 = 4$ .
- Initially, barrel  $B_1$  is filled with  $b_1$  litres of beer, barrel  $B_2$  is filled with  $b_2$  litres of beer, and barrel  $B_3$  is filled with  $b_3$  litres of beer.
- We want to decide whether or not it is possible to perform a sequence of steps that results in barrel  $B_1$  having  $b'_1$  liters of beer, barrel  $B_2$  having  $b'_2$  litres of beer, and barrel  $B_3$  having  $b'_3$  litres of beer?

(1.1) Formulate this as a problem on a directed graph. What are the vertices of the graph? What are the directed edges of the graph?

**Solution:** At any moment, the total amount of beer is equal to 4 litres. The current "state" is completely determined by the amount of beer in each barrel.

For any three integers  $b_1$ ,  $b_2$ , and  $b_3$  with  $b_1 \ge 0$ ,  $b_2 \ge 0$ ,  $0 \le b_3 \le 3$ , and  $b_1 + b_2 + b_3 = 4$ , there will be one vertex, which we denote by  $(b_1, b_2, b_3)$ .

How many vertices are there? I am sure you remember from COMP 2804 that the number of integer solutions to the equation  $b_1 + b_2 + b_3 = 4$  with  $b_1 \ge 0$ ,  $b_2 \ge 0$ , and  $b_3 \ge 0$ , is equal to

$$\binom{4+3-1}{3-1} = \binom{6}{2} = 15.$$

Among these 15 solutions, there is one that cannot occur, namely (0, 0, 4). Thus, our graph will have 14 vertices.

There is a directed edge from a source vertex to a target vertex if we can go in one step from the source vertex to the target vertex.

The decision problem becomes: given two vertices  $(b_1, b_2, b_3)$  and  $(b'_1, b'_2, b'_3)$ , is there a directed path from the first vertex to the second vertex.

(1.2) Draw the entire graph.

**Solution:** As was to be expected, this is a pain to do. The two figures below show the graph. The first figure only shows the directed edges (u, v), such that the reverse, i.e., (v, u) is not an edge. The second figure shows all "symmetric" edges, i.e., those edges (u, v) for which (v, u) is also an edge. I made two figures, because everything in one figure is a complete mess.



(1.3) Assume that  $(b_1, b_2, b_3) = (4, 0, 0)$  and  $(b'_1, b'_2, b'_3) = (3, 1, 0)$ . Use your graph to decide whether the answer to the decision problem is YES or NO.

**Solution:** We have to decide if there is a directed path from vertex (4, 0, 0) to vertex (3, 1, 0). The answer is YES. Here is one example of such a path:

$$(4,0,0) \to (0,4,0) \to (0,1,3) \to (3,1,0)$$

(1.4) Assume that  $(b_1, b_2, b_3) = (4, 0, 0)$  and  $(b'_1, b'_2, b'_3) = (2, 1, 1)$ . Use your graph to decide whether the answer to the decision problem is YES or NO.

**Solution:** We have to decide if there is a directed path from vertex (4, 0, 0) to vertex (2, 1, 1). Since the vertex (2, 1, 1) does not have any incoming edges, the answer is NO.

**Problem 2:** Let G = (V, E) be an undirected graph. A vertex coloring of G is a function  $f: V \to \{1, 2, ..., k\}$  such that for every edge  $\{u, v\}$  in E,  $f(u) \neq f(v)$ . In words, each vertex u gets a "color" f(u), from a set of k "colors", such that the two vertices of each edge have different colors.

Assume that the graph G has exactly one cycle with an odd number of vertices. (The graph may contain cycles with an even number of vertices.)

What is the smallest integer k such that a vertex coloring with k colors exists? As always, justify your answer.

**Solution:** Since the graph has an odd cycle, it cannot be colored using two colors. (This is the same as saying that the graph is not bipartite.)

We will show that the graph can be colored using three colors. Let  $(v_1, v_2, \ldots, v_k, v_1)$  be the unique odd cycle in G.

We remove one edge of this cycle, say  $\{v_1, v_2\}$  from G, and denote the resulting graph by G'. (We only remove this edge, we do not remove the vertices  $v_1$  and  $v_2$ .)

We observe that the graph G' does not contain any odd cycle. Therefore, as was mentioned in lecture 10, G' is bipartite. Thus, we can split the vertex set V into two sets, say B and R, such that every edge in G' has one vertex in B and the other vertex in R.

We give every vertex of B the color blue, and every vertex of R the color red. This is a valid vertex coloring of G' using two colors. However, the vertices  $v_1$  and  $v_2$  have the same color. Thus, if we add the edge  $\{v_1, v_2\}$  to G', we do not get a vertex coloring of the original graph G. We do get a vertex coloring of G, by giving  $v_1$  a new color, say green.

**Problem 3:** Let G = (V, E) be a directed graph, which is given to you in the adjacency list format. Thus, each vertex u has a list that stores all vertices of the set

$$\{v: (u,v) \in E\}.$$

The backwards graph  $G_b$  is obtained from G by replacing each edge (u, v) in G by the edge (v, u). In words, in  $G_b$ , we follow the edges of G backwards.

Describe an algorithm that computes, in O(|V| + |E|) time, an adjacency list representation of  $G_b$ . As always, justify your answer and the running time of your algorithm.

**Solution:** For each vertex u, we write its adjacency list in G as A(u), and we write its adjacency list in  $G_b$  as  $A_b(u)$ . The main observation is that for any two vertices u and v,

v is in A(u) if and only if u is in  $A_b(v)$ .

The algorithm does the following:

- For each vertex u, initialize an empty list  $A_b(u)$ .
- For each vertex *u*, do the following:
  - For each vertex v in A(u), add the vertex u to the list  $A_b(v)$ .

For the running time, initializing the lists  $A_b$  takes O(|V|) total time. The total time for the nested for-loops is proportional to

$$\sum_{u \in V} (1 + |A(u)|) = |V| + |E|.$$

Thus, the total time for the entire algorithm is O(|V| + |E|).