1. Create root vertex
2. Add root vertex to queue
3. While queue is not empty
   1. Take vertex off queue (V0)
   2. Parse its ID string ID[2]
   3. Set container-a curr\_volume to first element
   4. Set container-b current volume to second element
   5. If container-a is not full
      1. Fill container-a
      2. Create a vertex with container-a volume and V0.ID[1]
      3. Add an edge into the graph from V0 to new vertex
      4. Add new vertex to queue
   6. If container-b is not full
      1. Fill container-b
      2. Create a vertex with V0.ID[0] and container-b volume
      3. Add an edge into the graph from V0 to the new vertex
      4. Add new vertex to queue
   7. If canPour from A to B
   8. If canPour from B to A
      1. Do something
4. BFS function
   1. Establish queue
   2. Get root node from graph and put on queue
   3. While queue is not empty
      1. currentVertex = Take vertex off queue
      2. If isSolutionVertex(currentVertex)
         1. Done : Return currentVertex
      3. else
         1. get list of connections to currentVertex
         2. for each connection
            1. get its vertex
            2. place on queue
   4. at this point : queue is empty, no solution found
      1. throw an exception?-
5. Pour function (source, destination)
6. canPour function (source, destination)
   1. Source must not be empty
      1. If source is empty, return false
   2. Destination can’t be full
      1. If destination is full return false
   3. All other conditions, return true