**CS-254 Algorithms Assignment 1**

**1. What does the origin of the word Algorithms have to do with the origin of the word Algebra?**

An Arabic mathematician named Al-Khowarizmi is widely considered to be the founder of algebra and the ideas of simplification and reduction. Khowarizmi wrote a book about his findings and the title, in Latin, was Liber Algorismi de numero Indorum. In Latin, then, *algorismi* came to mean the process of breaking down a problem into all of the possible steps to find a robust solution. When it was translated to English, the word became *algorithms* as we know it today.

**2. Write down walking instructions for going from your seat in Algorithms classroom to your dorm, car, or apartment with the precision required from an algorithm’s description.**

* Stand up, slide chair backwards
* Take one step to the right of the chair
* Take coat from chair, put on
* Place any materials on the desk into backpack
* Put on Backpack
* Push in chair
* Face due West, walk until you have reached one step out from table edge
* Turn due North, walk until you are in the middle of the hallway
* Turn due East, walk until the descending stairs are due North from your position
* Turn due North, proceed down stairs until you are 2 steps from North wall
* Turn due East, walk until descending stairs are due South from your position
* Turn due South, proceed down stairs until you are 2 steps from South wall
* Turn due West, walk until West windows are 3 steps from your position
* Turn due North, walk until you reach the doors
* Open the first door, and walk through
* Continue North until you reach the second door
* Open the second door, and walk through
* Continue North until you have reached the grassy patch between the sidewalk and the road curb.
* Turn due West, walk until you have reached the sidewalk running parallel to 3rd street east on the East side of the street.
* Turn due North, ensure no vehicles are approaching from either the West or East direction on 10th Avenue E
* If no vehicles are approaching, cross to the other side of 10th Avenue E. If vehicles are approaching, wait until no vehicles are approaching
* Turn West. Ensure no vehicles are approaching from the North on 3rd street east. If no vehicles approaching; cross. If vehicles approaching, wait until no vehicles are approaching
* Continue walking West on the sidewalk North of 10th Avenue East until you reach the sidewalk running parallel to Broadways St S on the East side.
* Press the Crosswalk button. Wait 3 seconds for oncoming cars to stop, then cross to the sidewalk on the West side of Broadway St S
* Turn North. Walk North on the sidewalk until you reach the sidewalk on the South side of Main St. E.
* Turn West. Walk West on the sidewalk South of Main St E until you reach the sidewalk on the East side of 2nd Street W
* Turn North. Ensure no cars are approaching from the North or South on 2nd St West, or from the East on Main St. E. If no cars approaching, walk North to cross to the other side of Main St E. If cars are approaching, wait until no cars are approaching, then cross.
* Walk along the sidewalk on the East side of 2nd Street W until you reach the sidewalk on the South side of 6th avenue E
* Ensure no cars are approaching from either North or South on 2nd Street W or from the East on 6th Ave E. If no cars are approaching, cross north to the other side of 6th Ave E. If cars are approaching, wait until no cars are approaching, and cross.
* Walk along the sidewalk on the East side of 2nd Street W until you reach the sidewalk on the South side of 4th Ave W
* Ensure no cars are approaching from either North or South on 2nd Street W or from the East on 4th Ave W. If no cars are approaching, cross north to the other side of 4th Ave W. If cars are approaching, wait until no cars are approaching, and cross.
* Walk along the sidewalk on the East side of 2nd Street W until the large pedestrian crossing with the speed table is due West of your position. Turn West
* Check if cars are approaching from the North or South on 2nd St W. If no cars are approaching, cross West to the other side of 2nd St West. If cars are approaching, wait until no cars are approaching
* Walk West until you are 2 steps from the edge of the sidewalk.
* Turn North. Walk until you hit the first door of Red Cedar Hall
* Open the door, and walk through.
* Swipe your student ID at the card reader on the West wall of Red Cedar Hall
* In the span of 5 seconds, continue North until you reach the second door of Red Cedar hall, open it, and go through.
* Congratulations, you have made it into Red Cedar Hall!

**3. Design an algorithm in pseudocode to find all the common elements in two sorted lists of numbers. For example, {2, 5, 5, 5} and {2, 2, 3, 5, 5, 7} would result in {2, 5, 5}. What is the maximum number of comparisons of numbers the algorithm makes if the length of the lists are m and n, respectively? Be sure to state input and output expectations. Example Algorithm Pseudocode:**

Algorithm → comparison(A[0...n-1], B[0...n-1])

//This function would take in two arrays (A and B) and determine which of their //elements are common

//Input: arrays A[0...n-1] and B[0...n-1] of real integer numbers

//Output: arraylist C[0...n-1] of elements A and B have in common

starting\_pos = 0

**for** *i**elements in A* **do**

**for** *j elements in B, starting at* ***starting\_pos* do**

**if** A[i] == B[j] **do //***if we have a match*

C.push\_back(A[i]); //*Add match value to C*

*increment* ***starting\_pos;*** *//This avoids repeating the same number*

*break internal for loop; //We no longer have to search B for the current A[i]*

**else if** A[i] < B[j] **do //***The lists are sorted, we don’t need to search further*

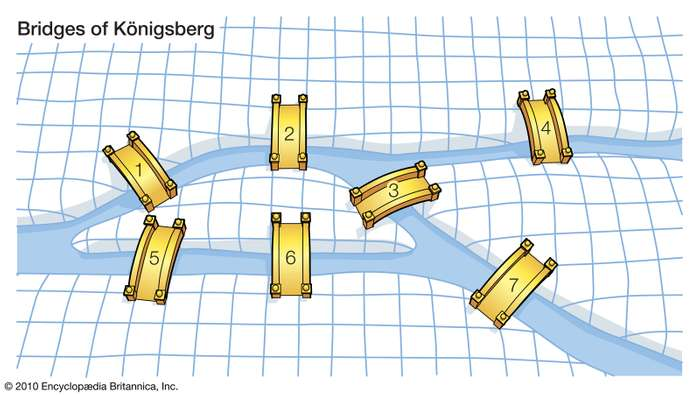
*decrement* ***starting\_pos; //****Back up one, we went too far*

*break internal for loop; //We no longer have to search B for the current A[i]*

**print(C);**

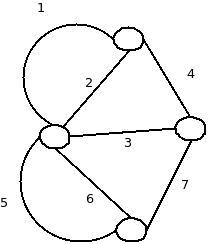
\*If arrays are of length m and n, the algorithm could end up making m\*n comparisons in the event that all elements in A are greater than all elements in B

**4. Assume you are making a new game and you need to cross all of the shown bridges, but once you cross a bridge you may never use the bridge again. Each of the land areas is a different world. You need to start and end in the same world. Can this be done in a single trip.**

****

**a) Try to state this game as a graph problem with nodes and edges.**

If we were to evaluate this game as a graph problem with nodes and edges, we could represent it with the following diagram.



Using the above diagram allows us to treat this as a graph theory problem. As such, to traverse each bridge once *and* return to our original world, we are finding an *Euler Cycle* (traverse each edge once and start and end at the same node).

**b) Draw the path of your trip or describe why you think it is not possible.**

Because we can treat this as a graph theory problem, we also now know that the scenario described is NOT POSSIBLE. For a graph to have an Euler Cycle, there must be no nodes in the graph that have an odd degree, that is, there are an odd number of edges on the node. In this case, we can clearly see that the North World, South World, and East Worlds all have a degree of 3, which means this graph does NOT have an Euler Cycle, and thus the described scenario of traversing each bridge once and starting and stopping in the same world is not possible.