Assignment 3

This assignment is worth 4% of your final grade. Answer all questions from 1 to 4. You may also answer the bonus questions for extra marks but you cannot receive partial marks for the bonus questions.

Please show steps of your reasoning. You may use any theorems (or lemmas or claims) that we proved in class if you clearly state which theorem you are using. Note that if a theorem you want to use is similar to (but not exactly) one seen in class, you should prove it first. Do not simply state that the proof is similar (write down the whole proof).

1. (a) (3 points) Give a polynomial time algorithm which takes as input a graph G with at least 3 vertices where all vertices have degree at least $|V(G)|/2$ and outputs a Hamiltonian cycle in G (that is, it outputs a sequence of vertices).

   Give your algorithm in pseudo-code and explain what each block of pseudo-code in your algorithm does. Do not put useless comments (i.e., comments that restate what a line of code does rather than stating your interpretation or expectation about that line of code).

   If you use unusual notation, do not forget to explain what it means.

   Hint: Finding the longest path in an input graph is NP-complete (being able to solve the longest path problem would allow us to solve the Hamiltonian path problem by checking if the longest path has length $|V(G)|$). However, you can still follow the proof of Dirac's theorem that we have seen for most of the steps of the algorithm.

   (b) (2 points) Prove that the algorithm you have given in a) runs in polynomial time. Write down an explicit value of k for which your algorithm runs in $O(n^k)$.

   (c) (5 points) Prove that your algorithm is correct. That is, show that what your algorithm outputs is a cycle and this cycle is Hamiltonian. (Depending on your algorithm, you may also need to show your algorithm does not get "stuck" before giving an output.)

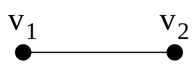
2.

   Definition 1. The *Cartesian product* of two graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$, denoted $G_1 \times G_2$, is a graph with vertex set V and edge set E defined as follows. V consists of all pair (v_1, v_2) for each vertex v_1 in V_1, and each vertex v_2 in V_2 (i.e., $V = \{(v_1, v_2)|v_1 \in V_1, v_2 \in V_2\}$). Two vertices (u_1, u_2) and (v_1, v_2) of $G_1 \times G_2$ are adjacent if either

   - $u_1 = v_1$ and u_2 is adjacent to v_2 in G_2, or
   - $u_2 = v_2$ and u_1 is adjacent to v_1 in G_1.

   In other words, $E = \{((u_1, u_2), (v_1, v_2))|u_1 = v_1, (u_2, v_2) \in E(G_2)\} \cup \{((u_1, v_1), (u_2, v_2))|u_2 = v_2, (u_1, v_1) \in E(G_1)\}$.

   (a) (3 points) Show that if G has a Hamiltonian cycle then $G \times P_2$ has a Hamiltonian cycle where P_2 is the graph on two vertices with a single edge (shown below).

   (b) (7 points) Show that if G_1 and G_2 both have Hamiltonian cycles and $|V(G_1)| = |V(G_2)|$ then $G_1 \times G_2$ has a Hamiltonian cycle.

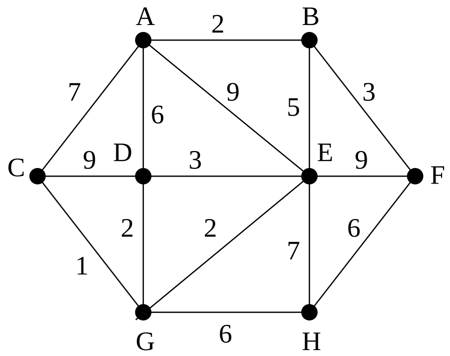
3. Prove or disprove each of the following statements. Depending on the negation of the statement, it may be sufficient to produce a counter-example (e.g., a graph) in case the statement is false.

   (a) (2 points) If T_1 and T_2 are trees then $|E(T_1)| = |E(T_2)|$.

   (b) (3 points) If $T = (V, E)$ is a tree then for any $e \in E$, $(V, E \setminus \{e\})$ is disconnected. That is, if we remove any edge from a tree, the resulting graph is disconnected.

(c) (5 points) If $T = (V, E)$ is a tree with at least 2 vertices then T has at least 2 vertices of degree 1. [Hint: Use induction and the lemma from class which already gives you one vertex of degree 1.]

4. (a) (3 points) Let G be the following graph.



Use Kruskal's algorithm to find the minimum spanning tree in G. Do not forget to show your steps (state in which order you are considering the edges and give a cycle for each unchosen edge).

(b) (7 points) Give a polynomial time algorithm which takes as input a connected graph $G = (V, E)$, weights $w_e \geq 0$ for each $e \in E$ and a subset of the edges $E_0 \subseteq E$ such that (V, E_0) contains no cycles. You algorithm should output a minimum weight subset F of the edges such that (V, F) contains no cycle and F contains E_0 (i.e., $E_0 \subseteq F$).

Prove that your algorithm indeed gives the minimum weight output. You may assume that your algorithm outputs a tree (although, you should make sure it does so) and that there is an optimal solution which is a tree.

5. **Bonus 1** (10 points) Prove the following statement.

   Let T be a tree. Any two longest paths in T share a common vertex.

6. **Bonus 2** (10 points) Implement the algorithm you gave in question 1 using any programming language. (Currently C, C++, Java, Perl and Python are supported, contact me in advance if you would like to use a different language. You may only use standard libraries.) You program should read from **stdin** and output to **stdout**.

   The **input** (graph) is specified as follows.

   The first line of the input contains a number $n \leq 1000$, the number of vertices. The vertices of the graph are assumed to be labelled from 1 to n. The subsequent n lines each contain a comma separated list of integers. The integers on the ith line corresponds to (the label of) all neighbours of the vertex labelled i.

   The **output** should be a space separated list of vertices (in the order of the Hamiltonian cycle).

   Your algorithm need not handle invalid inputs (i.e., it is allowed to crash or do anything if the input is not as specified, or is not a graph or is not a graph with the required properties.).

Sample input

```
5
2,3,5
3,4,5,1
4,1,2
5,2,3
1,2,4
```

Sample output

```
1 2 3 4 5
```

Here is the graph corresponding to the sample input with the Hamiltonian cycle of the output highlighted.