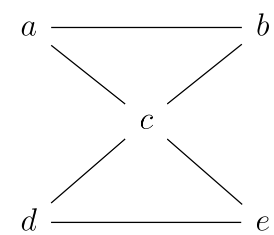
MA2C03: ASSIGNMENT 3
DUE BY WEDNESDAY, MARCH 1ST
SOLUTION SET

BRIAN TYRRELL

1) First we'll draw the graph:



(a) A graph is *connected* if and only if there exists a walk between every pair of vertices. By inspection we see this is true.

(b) The degree of every vertex is even, therefore the graph admits an Eulerian circuit. In particular, this is an Eulerian trail.

(c) The degree of every vertex is even, therefore the graph admits an Eulerian circuit.

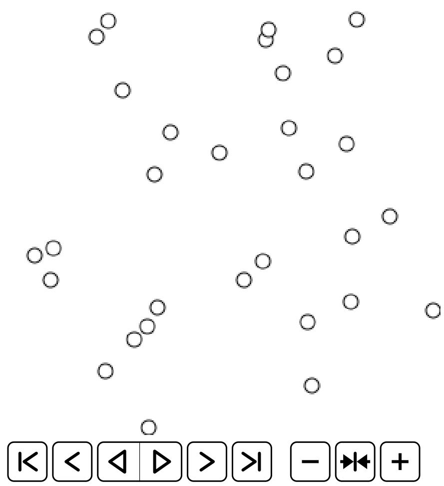
(d) No, there is no Hamiltonian circuit. For the purpose of contradiction, assume there is. The graph has horizontal an vertical symmetry, so without loss of generality we can assume the circuit starts (and thus ends) at a or c. If the circuit starts at a, to reach d and e and return to a it must pass through c twice; a contradiction. If the circuit starts at c, it must pass through c at least once to reach the vertices above and below c. Therefore we visit c twice; a contradiction. In either case there is no Hamiltonian circuit, as required.

(e) This graph has an Eulerian circuit (thus a circuit) ensuring it is not a tree.

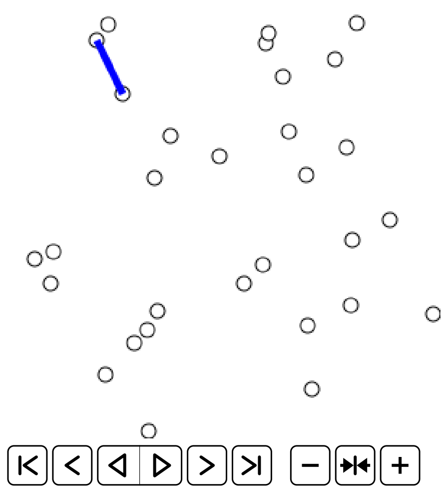
Grading rubric and common mistakes: 2 points for each part. Many people lost marks in (b) and (c) for not providing justification. You cannot give "the graph has an Eulerian trail because it does" as justification - you need a concrete reason, such as an example of an Eulerian trail (*acdecba*) or noting every vertex has even degree (and the graph is connected) leading to an Eulerian circuit.

2) See attached pages at the end. The Wikipedia page for Kruskal's algorithm and Prim's algorithm has gifs demonstrating the algorithm (here the weight of an edge is the distance between the two vertices of the edge). This pdf file needs to be opened in Adobe for the gifs to work.

Kruskal's algorithm:



Prim's algorithm:



Grading rubric and common mistakes: more specifically, for (b) and (c) 4 points were awarded for obtaining the correct minimal spanning tree, and 5 points for doing what the question asked, namely "draw the graph corresponding to each step of the algorithm". A lot of students did not do this. Also note (c) asks you to start with vertex

E, another thing a lot of students did not do. Finally, in Kruskal's algorithm your initial graph is $(V, \{\})$ and in Prim's your initial graph is $(\{E\}, \{\})$. Many students did not pay attention to this detail too.

3)

(a)

$$\begin{tikzcd}
A \arrow[r] \arrow[d] \arrow[loop left] & B \arrow[loop right] \\
C \arrow[loop left] & D \arrow[l] \arrow[u] \arrow[d] \arrow[dl] \\
F \arrow[r, bend right] & E \arrow[l]
\end{tikzcd}$$

(b) The adjacency matrix is as follows (note that we must stick to the given ordering of vertices; A, B, C, D, E, F):

$$\begin{pmatrix}
1 & 1 & 1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 \\
0 & 1 & 1 & 0 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 1 & 0
\end{pmatrix}$$

(c) With examination of the digraph we see B and C are so similar (two edges leading in and one loop) that we can try switch them in place, i.e.

$$\begin{cases}
\varphi(A) = A \\
\varphi(B) = C \\
\varphi(C) = B \\
\varphi(D) = D \\
\varphi(E) = E \\
\varphi(F) = F
\end{cases}$$

We just need to check this is an isomorphism of digraphs:

✓ φ is a bijection of vertices.

✓ All edges not involving B or C are unaffected, thus preserved. For edges involving B and C;

$$AB \mapsto AC \in \mathcal{E}$$
$$DB \mapsto DC \in \mathcal{E}$$
$$AC \mapsto AB \in \mathcal{E}$$
$$DC \mapsto DB \in \mathcal{E}$$

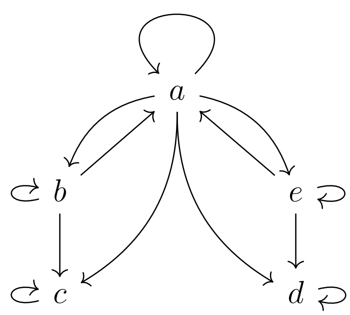
therefore the edge relation is preserved.

✓ Edge direction is also preserved by the above point.

Grading rubric and common mistakes: 2 points for (a) and 4 points each for (b) and (c). A number of students are loosing marks in Q2 (a), Q3 (a) and Q4 (a) for not drawing the correct graph, usually omitting edges. Please read the question carefully. A number of marks were also deducted for not having the correct adjacency matrix; the most common theme being students drawing an adjacency *table* for an *undirected* graph. The isomorphism question - (c) - was quite well answered, however.

4) This question contains a little twist at the end. First, we'll graph R:

(a)



(b) No; it is not symmetric - $(e, d) \in R$ however $(d, e) \notin R$.

(c) We would need to add the following pairs:

$$\text{To make } R \text{ symmetric:} \quad (c, b), (d, e), (c, a), (d, a)$$

$$\text{To make } R \text{ transitive:} \quad (b, e), (e, b), (b, d), (d, b), (c, e), (e, c), (c, d), (d, c)$$

(d) In order to be reflexive R' needs 5 elements;

$$R' = \{(a, a), (b, b), (c, c), (d, d), (e, e)\}$$

(e), (f) Recall the definitions for the properties of an equivalence relation R':

(1) *Reflexive:* $\forall x(xR'x)$.
(2) *Symmetric:* $\forall x, y(xR'y \to yR'x)$.
(3) *Transitive:* $\forall x, y, z(xR'y \wedge yR'z \to xR'z)$

The minimum number required for (e) and (f) is **zero**, i.e. $R' = \emptyset$. This may seem surprising, but these relations hold *vacuously.* I spoke briefly about vacuous truth at the start of the year, where I offered the following example during our tutorial: *all elephants in this room are purple.*

Of course, there were no elephants in the room at the time, so the statement was *vacuously true* - think of it as 'true because it's not wrong'. In order for the statement *all elephants in this room are purple* to be false, you would have to show there was a non purple elephant in the room, which could not be done.

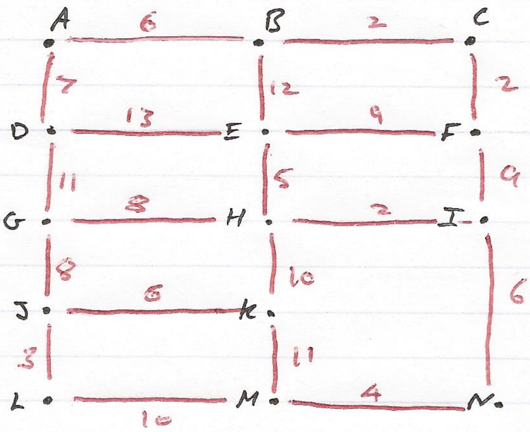
Similarly, in order for symmetry to be false for $R' = \emptyset$, there needs to be an element contradicting symmetry in R' i.e.

$$\neg(\forall x, y(xR'y \to yR'x)) \qquad \leftrightarrow \qquad \exists x, y(xR'y \wedge \neg(yR'x)).$$

However, since R' is empty, it has no elements to contradict symmetry, thus it *is* symmetric. Similarly, $R' = \emptyset$ is transitive, meaning the minimum number of ordered pairs required for (e) and (f) is zero.

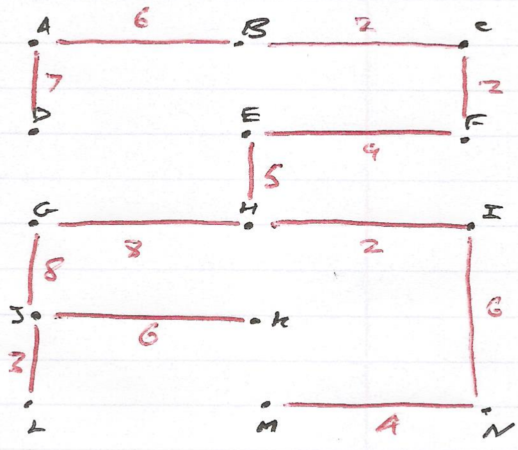
Grading rubric and common mistakes: 2 points for (a), 2 points for (b) and 4 points each for everything else. There were twelve vertices needed for (c) and four points available - students were awarded points relative to how many vertices they put down. I'd wager the majority of the class got (d), (e) and (f) correct, however some justification for your answer is needed (I was generous for this one).

2) a)



End result:

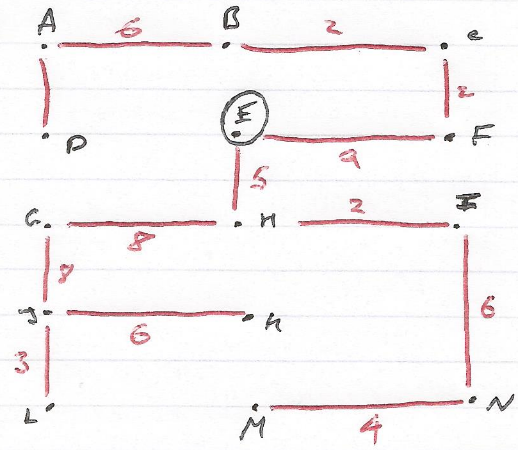
b)



BC → EF added.
Rest are discarded.

End result:

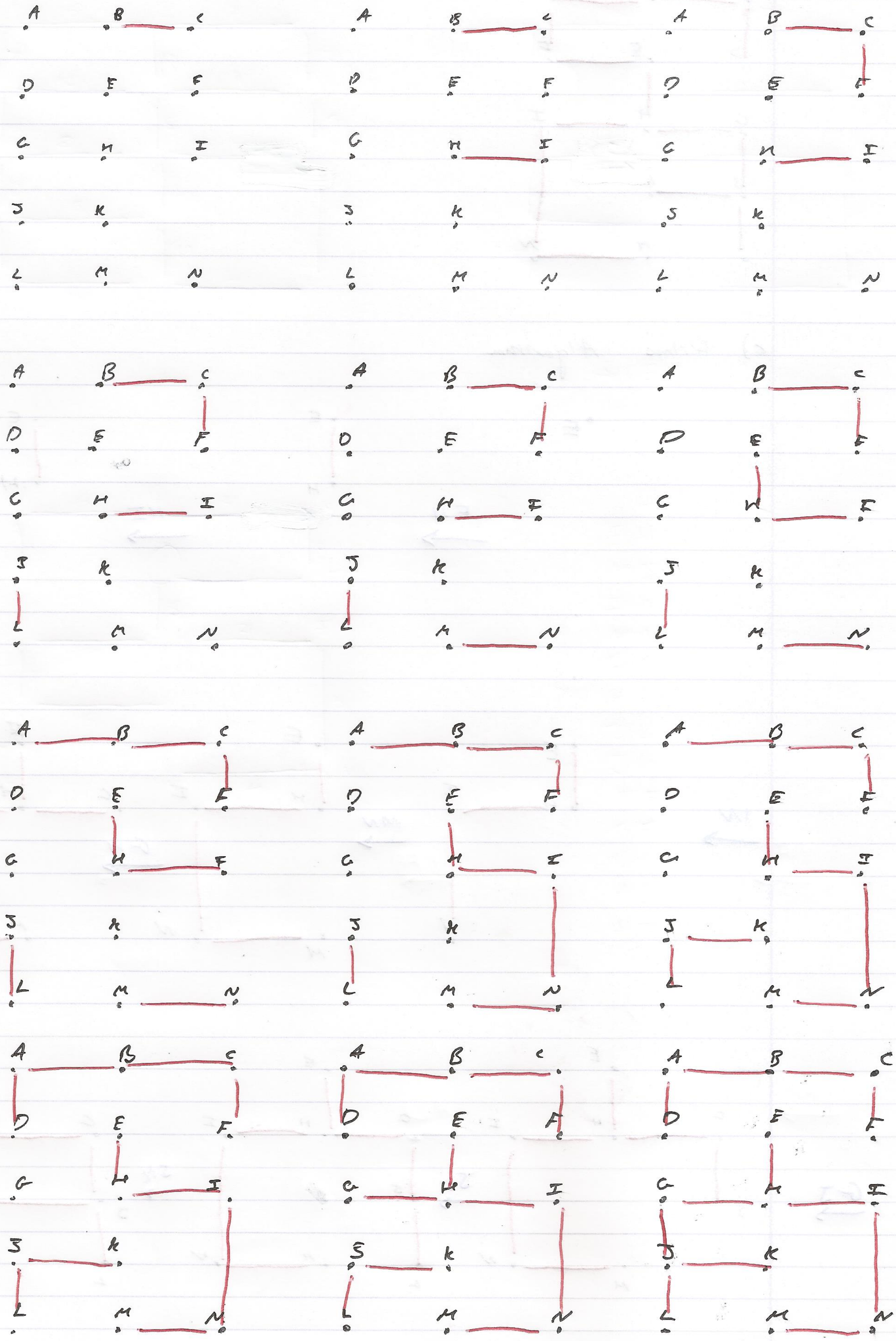
c)

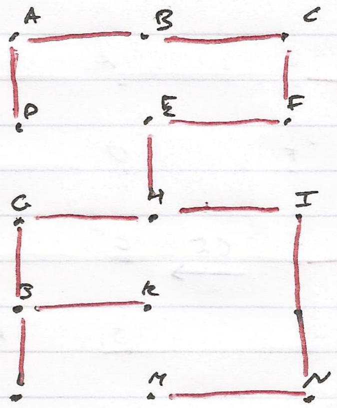


Add: EH
HI
IN
MN
GH
GJ
JL
JK
EF
CF
BC
AB
AD

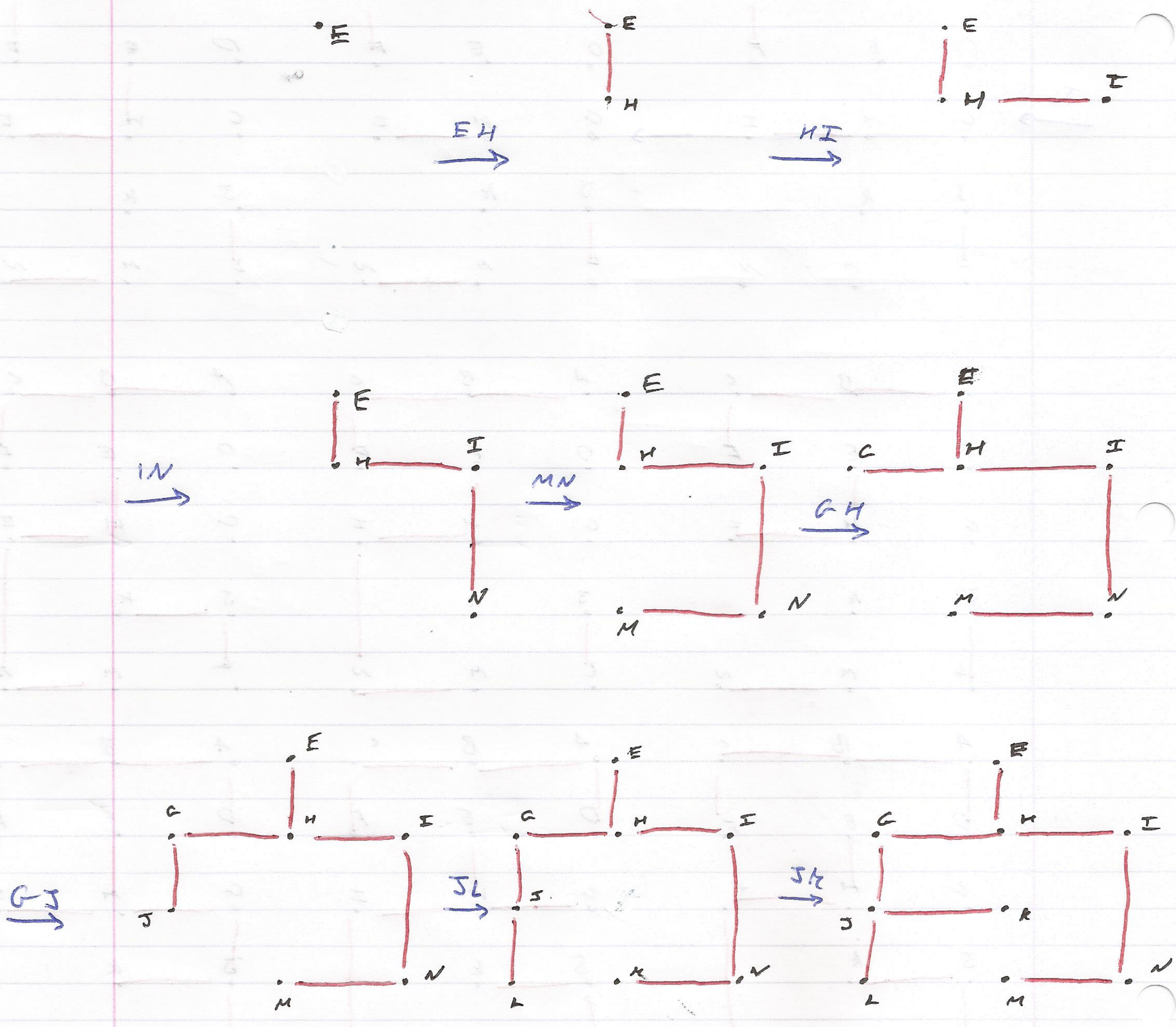
Kruskal's Algorithm

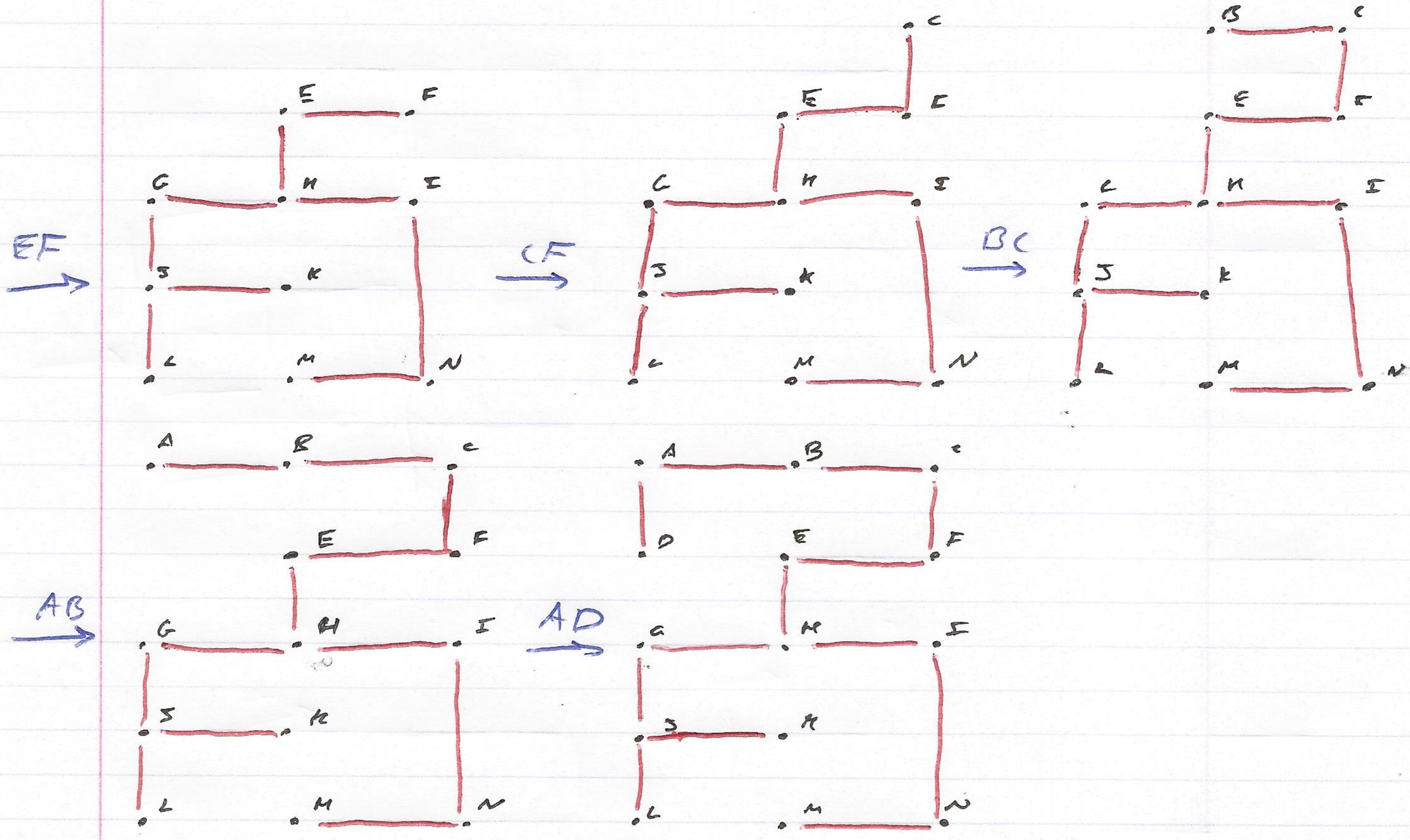
b)





c) Prim's Algorithm




EF
CF
BC
AB
AD