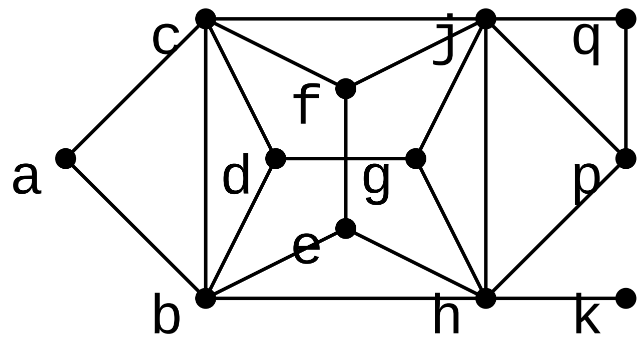
cmput 304 2023 study questions 7: part 2

1. Recall: for a graph $G = (V, E)$ and node subset $U \subseteq V$, $G[U]$ is the subgraph of G induced by U (all the nodes of U together with all edges of G that have both ends in U). For the graph G below, let $H = (X, F) = G[\{a, c, e, g, j, p, q\}$. Give X (the node set of H) and F (the edge set of H).



2. Recall: a *clique* is a graph $G = (V, E)$ is a subset K of V, such that for each pair of nodes x, y in K, (x, y) is in E. Recall: a clique in a graph is *maximal* if it is not a proper subset of any larger clique. For example, in the graph above $\{a, c\}$ is not-maximal clique and $\{a, b, c\}$ is a maximal clique.

   (a) List all maximal cliques of the above graph G.

   (b) List all maximal independent sets of $G[\{a, b, c, d, e, f, g, j, h\}]$.

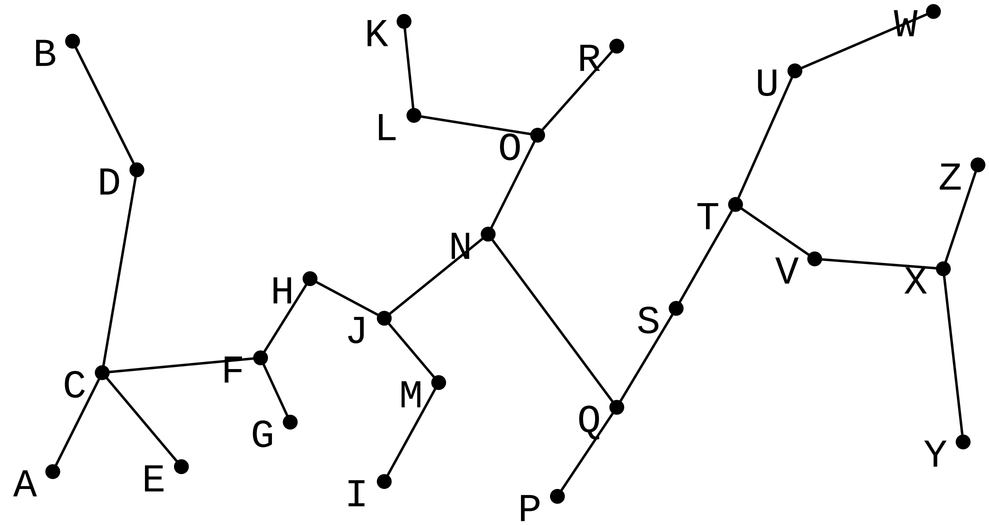
3. Recall: a node is *simplicial* in a graph if its set of neighbors is a clique.

   (a) List all the simplicial nodes in the graph in question 1.

   (b) Let v be simplicial node in a graph G. Let M be the set of non-neighbors of v. Let I_M be a largest independent set in $G[M]$. Prove that $\{v\} \cup I_M$ is an independent set in G.

   (c) Prove that $\{v\} \cup I_M$ is a largest independent set in G.

4. For the graph G below, find a largest independent set in $G[\{A, B, C, \ldots R\}$.



5. Recall: a *forest* is a graph with no cycles, a *tree* is a connected forest, and a *leaf* is a node with at most one neighbor.

(a) Prove that every leaf in a forest is simplicial.

```
input: a tree T
I <- { } % empty set
while T has at least one node:
  remove from T any leaf v
  add v to I
  if v has a neighbor w:
    remove v, w, and edges (v,w) from T
  else:
    remove v from T
return I
```

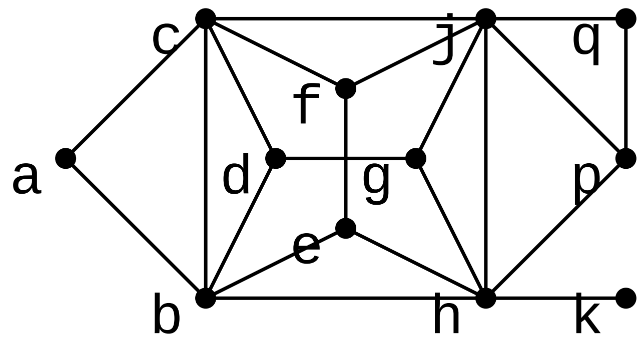
(b) At each execution of the while loop, is T a tree (i.e. connected and acyclic)?

(c) For any tree T, is the set I returned by the algorithm a largest independent set of T?

(d) In the algorithm, if T is initially connected but has a cycle (so not a tree), will the algorithm still always work? If yes, explain. If no, give an input graph where the algorithm does not work.

(e) In the algorithm, if T is initially acyclic but disconnected (so not a tree), will the algorithm still always work? If yes, explain. If no, give an input graph where the algorithm does not work.

6.



Let G be above graph. Let H be the graph obtained from G by removing nodes in $X = \{k, p, q, \}$ (and all edges that include an edge with a node in X).

(a) Formulate the problem of finding a largest independent set of H as an integer program.

(b) If you solve (a) using the mixed-integer-program solver at sagemath, will the solution be all integer, or might there be some non-integer (fractional) values in the solution? Explain.

hints

1. $X = \{a, c, e, g, j, p, q\}$. $F = \{(a, c), (c, j), (g, j), (j, p), (j, q), (p, q)\}$.

2. (a) `https://webdocs.cs.ualberta.ca/~hayward/304/asn/ISperf.pdf` see page 3.
   (b) $\{a, d, e, j\}$, $\{a, d, f, h\}$, $\{a, d, e, g\}$, $\{a, e, j\}$, $\{a, f, h\}$.

3. (a) a, k, p

   (b) Rename $\{v\} \cup I_M$ as I'. Let x, y be any two nodes in I'. Then those two nodes are not adjacent. (Why? If one of the nodes is v, then the other is in the non-neighborhood of v and they are not adjacent. If neither node is v, then both are in I_M, which is an independent set, so they are not adjacent.)

   (c) Let I^+ be any largest independent set in G. Let I^+_M be the subgraph of I^+ restricted to the set M. Claim: I^+_M is a largest independent set of $G[M]$.

   Prove the claim by contradiction: assume that I^+_M is not a largest independent set of $G[M]$. Then there is some Z that is larger than I^+_M and an independent set of $G[M]$. Then $Z \cup \{v\}$ is an independent set of G (why? because v is non-adjacent with every node of M) and larger that $I^+ = I^+_M \cup \{v\} = I^+$, contradicting the assumption that I^+ is a largest independent set of G. So the claim holds.

   Now I^+ and $\{v\} \cup I_M$ are both independent sets and have the same size. I^+ is a maximum independent set of G, so $\{v\} \cup I_M$ is also a largest independent set of G.

4. There are many correct answers. If you run the algorithm from question 5 on this graph and assume that the node list is in alphabetic order, then you get $\{A, B, E, G, H, I, K, N, P, R\}$.

5. (a) The set of neighbors of a leaf is either the empty set (so, vacuously a clique) or has exactly one element (again, vacuously a clique).

   (b) No. For the graph in question 4, if the algorithm first removes leaf A and its neighbor C, then the remain graph has 3 components: $\{E\}$, $\{B, D\}$, and the rest of the nodes.

   (c) Yes, by (a) and question 3.

   (d) No. If the input graph is a triangle, the algorithm will fail to find a leaf, and so not terminate.

   (e) Yes. The input is a forest. Every forest with at least one node has a leaf. Removing a leaf and its neighbor, if it has one, leaves a forest.

6. (a) From page 3 of

`https://webdocs.cs.ualberta.ca/~hayward/304/asn/ISperf.pdf` :

```
nodes x_1 .. x_12      max'l cliques  y_1 .. y_10
  abc bcd beh cfj  dg  ef ghj  hk hjp jpq
max [1 1 .. 1] X s.t. AX <= [1 1 .. 1]^t
         a b c d e f g h j k p q
    A   |1 1 1                  |
        |  1 1 1                |
        |  1     1     1        |
        |    1     1     1      |
        |      1     1          |
        |        1 1            |
        |            1 1 1      |
        |              1   1    |
        |              1 1   1  |
        |                1   1 1|
```

Remove nodes in $X = \{k, p, q\}$, remove maximal cliques containing any node in X, and add cliques that are now maximal but weren't before.

(b) In class, I mentioned that this graph is from a class of graphs called perfect, that have this property: the max IS (independent set) IP (integer program) formulation has all extreme points all-integer, so any LP-solver that uses the simplex algorithm (like sagemath) will always find an all-integer solution to the max IS IP problem.