**Fall 2022**

**Derek Stearns**

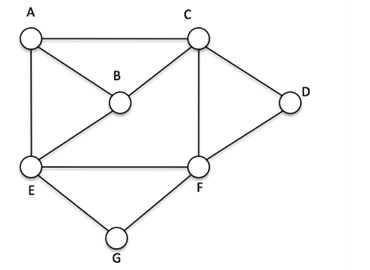
**CSCI 4150/8156**

**Homework Three**

1. (5 pts) A city planning engineer would like to determine the minimum number of security cameras it would take to cover all streets in a high-profile area. The cameras will be placed at some corners so that every street is covered. A street is covered if there is a camera at either end of the street. An example of such a network of streets is shown below.
   1. (3 pts) Describe a model of the city to solve this problem. What graph problem can be used to solve this problem? For the example below, what is the minimum number of needed cameras?

The cameras can be nodes of a graph. Edges can represent streets so that the minimum edge covering of the graph G could represent where a camera would need to be placed to see each street. In this case it would be our Maximum matching of |3 – n| = 4. The nodes that provide the minimum edge cover are C, E, A, F.

* 1. (2 pts) Provide two different examples of neighborhood areas for which two cameras will be enough to cover all streets. Provide an example for which at least three cameras are needed.



An example where at least two cameras would be needed at B and F:

Chart

Description automatically generated

An example where at least 3 cameras are needed (B,F,G):

Chart

Description automatically generated

1. (5 pts) For the task graph *G* in the Figure below, answer the following:
   1. (3 pts) Obtain the complement graph *G'* and find the parameters α(*G’*), α’(*G’*), β(*G’*), β’(*G’*), and γ(*G’*).

α(*G’*): 3 (E,D,B)

α*’* (*G*): 3 (A,B,G)

β(*G’*): 2 (C,F)

β’(*G’*): 4 {(A,B),(C,D),(G,F),(F,E)}

γ(*G’*): 2

Chart

Description automatically generated

* 1. (2 pts) Find a maximum matching *M* in the graph *G’* that corresponds to an optimal twoProcessor schedule of *G* and verify the scheduling-matching relationship: *S = n - |M|*.

The maximum matching of G’ is 3 {(A,B),(C*,*D),(G,E)}

A schedule length “S” is equal to the number of tasks – Matching in G’ where the number of tasks is S = 7 – 3 = 4, which follows logic that the minimum possible schedule (not including direction) would be:

The Modulus of The number of tasks / Number of processors : 4

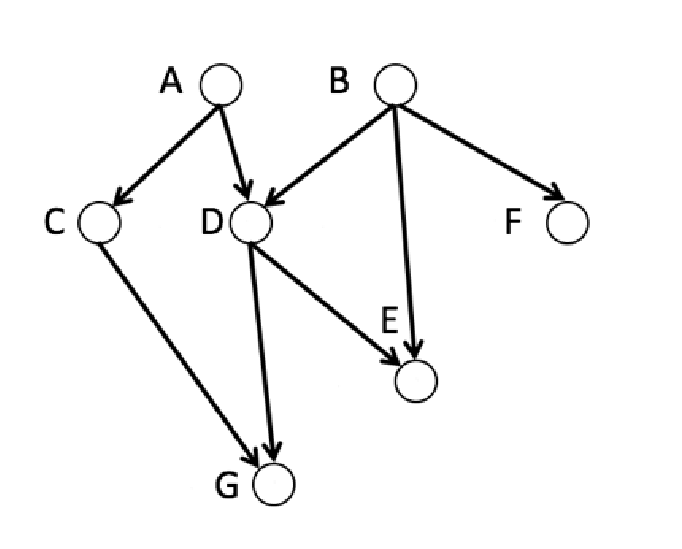
Where the schedule of G with two Processors is:

A,B

C,D

G,E

F



1. (2 pts) Let *X* be a set of plants, and let *Y* be a set of different types of soil, find necessary and sufficient conditions such that each plant in the set *X* can be planted in at least *k* soils that are suitable for its growth. You can assume that for each type of plant and each type of soil, the relationship is either suitable for growth or not suitable for growth.

The condition that there is a valid match M for a set of plants X and a set of soils Y must follow that all the possible sets of X in graph G follow Hall’s condition where the neighborhood of soils from set X, or all soils included in the adjacent set in X, is greater than or equal to the number of plants in a set X. The graph must also be bipartite (which is logical that a plant cannot be planted in a plant nor a soil planted in soil to yield suitable growth).

1. (4 pts) The following is a simple heuristic for finding a vertex cover. In each step of the algorithm, the vertex of the highest degree (ties broken arbitrarily) is added to the cover, then it is removed from the graph together with all its incident edges. The algorithm terminates only when no more edges remain. Show an example of a graph and an order of execution of this algorithm that leads to a minimum vertex cover and another for which the obtained cover is not minimum.

For a graph G = (V, E)

While (E is not empty) { add vertex v with max deg(v) to the set V’ where deg(v)=degree of vertex v and V’ is the vertex cover

Remove all edges incident to v from E}

return V’

The minimum vertex cover in the other graph is found by node set {F, B, C, G}

Chart

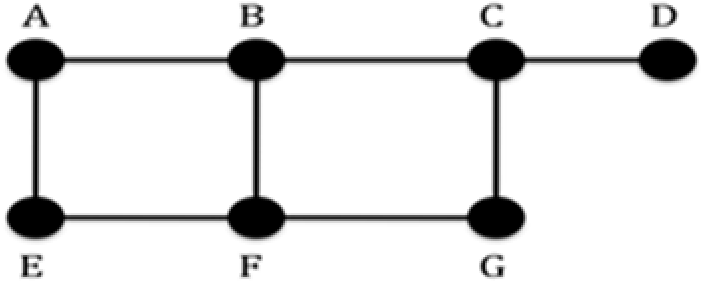
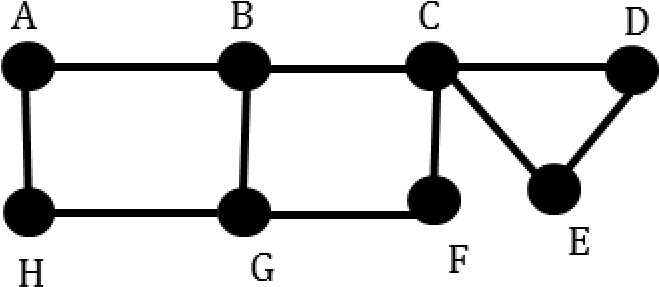
Description automatically generated

The algorithm would not find a minimum vertex cover in {A,G,F,E} as the MVC is {G,F,E}

Chart

Description automatically generated

1. (4 pts) Apply the matching algorithm on each the following graphs. Show your steps.

Since the first graph is bipartite, we can use an alternating path algorithm by the following steps:

1. Choose an edge e.g. (E,F) to be in the matching
2. (E,F) also has adjacent edges not in the matching (A,E), (F,G)
3. The edges not in the matching constitute an augmenting path.
4. Taking the edges not in the matching as the new matching (A,E) (F,G) and continuing the path by adding an edge not in the matching (G,C) and adding a final edge to the matching (C,D) we find a maximum matching of 3

Since the second graph has an odd cycle we must:

1. First collapse the cycle to a point so C becomes a new vertex
2. Find a matching by alternating paths (A,B),(H,G),(C,F)
3. Uncollapse the nodes of the blossom at C and add the remaining alternating path edges where in this case (D,E) is added to the path (A,B),(H,G),(C,F) for a maximum matching of 4.

Due on Wednesday 10/12/2022