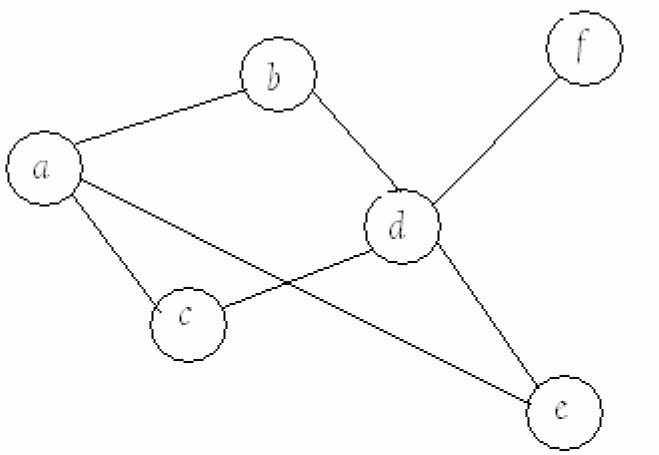
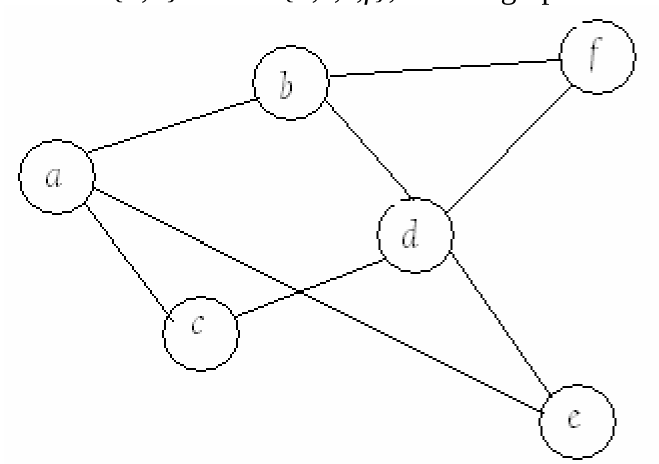
CS2223
HW#8

DUE: Monday, May 1

1. (10 points) A graph $G=(V,E)$ is *bipartite* if its vertices *V* can be partitioned into two classes, *A* and *B*, such that every $e \in E$ joins a vertex from *A* with a vertex from *B*. That is, there does not exist an $e \in E$ joining two vertices in *A* or two vertices in *B*. For example, the graph



is bipartite with classes *A*={*a*,*d*} and *B*={*b*,*c*,*e*,*f*}, but the graph



is not bipartite. It can be shown that a graph is bipartite if and only if it does not admit any cycles of odd length. The second graph above admits a triangle, with vertices *b*, *d* and *f*, and hence it is not bipartite. Describe an algorithm to test if a graph is bipartite. The worst case execution time of your algorithm should be in $O(|V|+|E|)$.

2. (8 points) Test the algorithm for estimating the size of a set developed in class by writing a program and testing it on a set S of 100 elements. Try several runs of your program and describe the estimates it gives for $|S|$.

CS2223
Solutions for HW#8

1. G is visited with enough *breadth first search*es to exhaust V (until every $v \in V$ is visited). This can be done in time in $O(|V| + |E|)$. For each bfs, the vertices at levels 0 (the root), 2, 4,... (the even levels) belong to one class, *A*, and the vertices at levels, 1, 3, 5,... (the odd levels) belong to the other class, *B*. The graph G is **not** bipartite if and only if it admits an odd cycle which is true if and only if the bfs-forest admits an edge with respect to a bfs-tree between a vertex at an even level (a vertex of *A*) and a vertex of an odd level (a vertex of *B*).