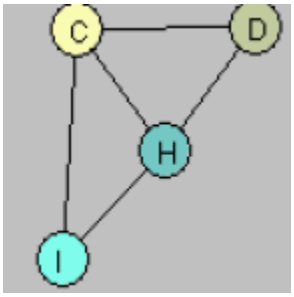
Final Exam Review Questions for Graphs

1. Name two applications for undirected graphs, and two more applications for directed graphs.
2. Among the edges of an undirected graph G are: (A,B) and (B,C). Consider the path p: A, B, C, B, A. Is p a cycle? Explain.
3. Suppose an undirected graph has 10 vertices. What is the maximum number of edges such a graph could have?
4. Suppose an undirected graph has 10 vertices, v_1, v_2, . . ., v_{10}, and 20 edges. Joe is counting how many edges come out of each vertex. He discovers that there are d_1 edges incident to v_1, d_2 edges incident to v_2, . . ., d_{10} edges incident to v_{10}. What is the sum $d_1 + d_2 + \ldots + d_{10}$?
5. Suppose an undirected graph G has 10 vertices and 75 edges. Must G be connected? Explain.
6. For the graph G below, if W = {C, H, I}, draw the graph G[W].



7. If an undirected graph has n vertices and n-1 edges, must it be a tree? If so, prove your answer. If not, give an example to prove your point.
8. Suppose T is a tree with n vertices. How many different *rooted* trees can be identified using T (without adding or removing any vertices or edges)?
9. Is there a connected (undirected) graph having n vertices and n-2 edges? Explain
10. Does every undirected graph have a vertex cover? Explain.
11. Is it possible for a graph with 11 vertices and 10 edges to have a vertex cover consisting of only one vertex?
12. What is an adjacency list used for?
13. Suppose G is an undirected graph and v is a vertex in V. How many times is v examined during BFS?
14. Why is it helpful, when implementing graphs in code, to represent the BFS algorithm as a separate class? In what way is this a useful approach?
15. Suppose G is an undirected graph having exactly two connected components. One of the components has n/3 vertices, the other has 2n/3 vertices (assume n is a multiple of 3). Assume also that n < m, where m is the number of edges. What is the asymptotic running time to perform DFS on G?

16. Suppose you are given a connected graph G, with vertex set V and edge set E. Give an O(1) algorithm for determining whether G contains a cycle.
17. A *discrete graph* is a graph that has no edges. What is the smallest size of a vertex cover for a discrete graph?
18. Explain why Dijkstra's Algorithm does not work when it is used on an undirected graph with one or more negative edge weights.
19. Suppose $v_1, v_2, v_3, \ldots, v_m$ is a shortest path from v_1 to v_m in a weigthed undirected graph. Explain why $v_1, v_2, v_3, \ldots, v_{m-1}$ must be a shortest path from v_1 to v_{m-1}.
20. Explain in a simple way what Kruskal's algorithm does. What is the input to the algorithm? Does Kruskal's algorithm work if there are negative edge weights? What is the output of the algorithm? Without using any kind of optimizations (in particular, without implementing a DisjointSets data structure), what is the running time of Kruskal's algorithm?
21. Explain why Kruskal's algorithm is an example of a greedy algorithm. Where precisely in the algorithm are greedy decisions made?
22. What is the DisjointSets data structure? What data does it store? What are its primary operations? What is the running time of each? What is this data structure used for in general? (Do not mention Kruskal's algorithm.)
23. Suppose U = {1,3,5,7, 9, 10} and C = {{1,3}, {5}, {7,9,10}}. Express these sets as trees (as was done in class). Then show how these trees are modified if we form the union of {1,3} and {7,9,10}.
24. In Kruskal's algorithm, the first step is to arrange the edges in increasing order of edge weight. Why is this done? What if this step were not taken? How would this affect the output?
25. In the optimized version of Kruskal's algorithm given in class, the DisjointSets data structure is used. How is it used? What problem does it solve? How exactly does this data structure improve the performance of Kruskal's algorithm?
26. In a weighted undirected graph G, if T_1 and T_2 are disjoint trees why must it be true that if there is an edge (x,y) in G where x is in T_1 and y is in T_2, then the union $T_1 \cup T_2 \cup \{(x,y)\}$ is also a tree?