CS146
Sample Final Exam KEY

1. Which of the following problems has no polynomial time algorithm
   a. Minimum spanning tree
   b. Shortest simple path in a graph
   c. <u>Longest simple path in a graph</u>
   d. Fraction Knapsack

2. Which of the following basic algorithms can be used to most efficiently determine the number of connected components in a graph:
   a. Kruskal's algorithm
   b. Bellman and Ford's algorithm
   c. <u>Depth-first search algorithm</u>
   d. Breadth First search algorithm

3. A problem is said to be NP-Complete
   a. A non-polynomial time algorithm has been discovered
   b. A polynomial time algorithm can exist but needs a parallel computer
   c. There is Greedy solution to the problem
   d. <u>If it is as 'hard' as any problem in NP</u>

4. What do you know about P=NP
   a. It is true
   b. It is false
   c. <u>It is open</u>
   d. a and c

5. Which of the following is not a technique for designing algorithms
   a. greedy
   b. exhaustive search
   c. divide and conquer
   d. dynamic programming
   e. <u>bouncing</u>

6. If you want to prove that problem A is NP-hard which of the following is the correct way to prove it
   a. $Shortest\ Path \leq_p A$
   b. $A \leq_p 3 - SAT$
   c. $\boldsymbol{LongestPath \leq_p A}$ **<u>(known NP-complete HC is no harder than A)</u>**
   d. None of the above

7. Consider the 0-1 knapsack problem. What is the best solution, with 4 items and W=10, such that w=(6, 3, 4, 2) –*weights*- and b=($30, $16, $28, $18) -*values*
   a. 76
   b. 68
   c. <u>62</u>
   d. none

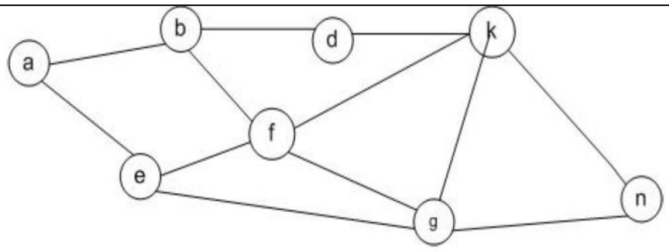
8. The distance matrix of a (directed) graph with vertices x, y, u and v is given by the shortest path from x to u consists of edges

	X	Y	U	V
X	0	7	∞	2
Y	8	0	10	5
U	3	∞	0	2
V	10	5	3	0

   a. xu
   b. xy and yu
   c. <u>xv and vu</u>
   d. there is no path

9. Which of the following problems is known to have a polynomial time solution
   a. Longest simple path problem for a given graph
   b. The 3-colorability problem in graphs
   c. <u>The Euler cycle in a graph</u>
   d. The Hamiltonian Cycle in a graph

10. For the following graph what is true *(note that nodes b and e have odd degree=> no Euler cycle)*:

a. <u>It has a Hamiltonian cycle</u> b. It has an Euler cycle c. a and b d. None	

11. Let G = (V, E) be a connected, undirected graph with edge weights w : E → Z. Suppose G has a unique minimum spanning tree. What can you conclude about G?
a. G contains no cycles
b. G contains at most one cycle
c. All edge weights are different
<u>d. None of the above</u>

12. If there exists an NP-complete problem which is in P then P = NP
a. <u>True</u>
b. False

13. state whether the following statement is true or false for Single Source Shortest Paths
    i. The problem is to determine the shortest paths from a source vertex s to all the remaining vertices of G.

        ii. The way to generate the shortest paths from s to the remaining vertices is to use relaxation
        iii. it is assumed that all the weights are positive
        iv. graph is complete
    b. True, true, true, false
    c. True, false, true, true
    d. <u>True, true, false, false</u>
    e. False, true, false, true

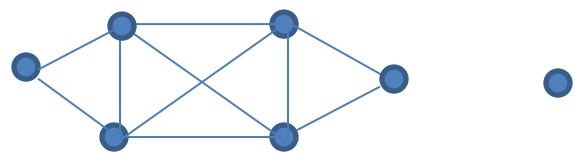
14. state whether the following statement is true or false for Spanning Trees
        i. A tree T is said to be a spanning tree of a connected graph G if T is a sub graph of G and T contains all the vertices of G
        ii. The minimum spanning tree is unique
        iii. Spanning is defined only for a connected graph
    a. True, true, true
    b. <u>True, false, true</u>
    c. True, false, false
    d. False, true, true

15. Which of the following problems has no polynomial time algorithm
    a. Sorting
    b. Shortest simple path in a graph
    c. <u>Hamiltonian cycle in a graph</u>
    d. Graph Searching

16. Which of the following basic algorithms can be used to most efficiently determine the presence of a cycle in a given graph
    a. Prim's algorithm
    b. Bellman and Ford's algorithm
    c. Floyd and Warshal's algorithm
    d. <u>Depth-first search algorithm</u>

17. A problem is said to be NP-Complete
    a. A non-polynomial time algorithm has been discovered
    b. A polynomial time algorithm can exist but needs a parallel computer
    c. There is Greedy solution to the problem
    d. <u>If it is as 'hard' as any problem in NP</u>

18. For the following graph what is true:

a. It has a Hamiltonian cycle **b. <u>It has an Euler cycle</u>** c. a and b d. None	

19. What do you know about P=NP
    a. It is true

b. It is false
c. <u>It is open</u>
d. a and c

20. We employ greedy approach when
    a. It gives optimal solution
    b. <u>The solution has optimal substructure</u>
    c. It is faster than dynamic programming
    d. The problem is NP-complete

21. What is the asymptotic complexity of $T(n) = 3T(\frac{n}{2}) + n^2$:
    a. $\Theta(n^2 \lg n)$
    b. $\Theta(\lg n)$
    c. <u>$\boldsymbol{\Theta(n^2)}$</u>
    d. None of the above

22. Which problem can be solved optimal by a dynamic programming algorithm
    a. Sorting
    b. Single source shortest path
    c. <u>All-pairs shortest path (Floyd-Warshal)</u>
    d. Clique

23. If you want to prove that problem A is NP-hard which of the following is the correct way to prove it
    a. <u>$\boldsymbol{Vertex\ Cover \leq_p A}$ (known NP-complete VC is no harder than A)</u>
    b. $Shortest\ Path \leq_p A$
    c. $A \leq_p 3 - SAT$
    d. None of the above

24. The array 1 4 2 5 8 6 10 is organized into a min heap (priority queue). Which array represents the heap after two deleteMin operations have been performed?
    a. 10 8 6 5 4
    b. 4 6 5 10 8
    c. <u>4 5 6 10 8</u>
    d. 4 10 6 5 8

25. Consider the 0-1 knapsack problem. What is a good example, with 3 items and W=50, that shows that being greedy does not provide the optimum profit
    a. w=(10,20,30) and b=(40,100,150)
    b. w=(10,20,30) and b=(70,100,60)
    c. <u>w=(10,20,30) and b=(60,100,120)</u>
    d. none

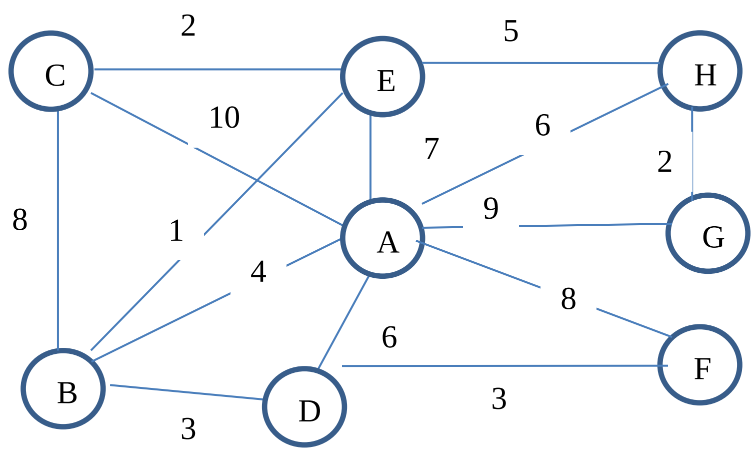
26. The (log n) -th smallest number of n unsorted numbers can be determined in
    a. O(nlogn) worst-case time
    b. <u>O(n) worst-case time (select algorithm – order statistic problem)</u>
    c. O(logn) worst-case time
    d. $O(n^2)$ worst-case time

27. You are driving alone from San Jose to Las Vegas in a Porsche 911 Carrera on a fixed 530 mile route. Assume that you can go 35 miles on one tank of gas and that you have a map showing every gas station on the route. What methodology for designing efficient algorithms would you choose to determine which gas stations to stop at and how much gas to get at each station?

a. Divide and conquer
b. **Greedy**
c. Dynamic programming
d. None of the above

SECTION II – Short Answer– Write a short answer to each question. [20 points each]

28. LuxuryForAll Construction is in the process of installing power lines to a large housing development. The owner wants to minimize the total length of wire used, which will minimize her costs. The housing development is shown as a graph in the next figure. Each house has been numbered, and the distance between the houses are given in hundreds of feet. What do you recommend? (Total length of wires and also which will be installed.) How would you solve it efficiently? Show all the steps.



Solution:
This is minimum spanning tree. Use Prim or Kruskal.
Prim start with vertex a choose AB, then BE, EC, BD, DF, EH, HG
Kruskal, BE, HG, EC, BD, DF, BA and EH

29. Suppose you must take n specified courses to earn a college degree, and each course has a set of required prerequisite courses. Design an efficient greedy algorithm that determines the fewest number of semesters that would be needed to take all n courses, assuming that there is no limit to the number of courses you can take simultaneously. What is the running time?

Answer:

```
S ← {all n courses};
semester ← 0;
while (S ≠ empty set) {
    semester++;
    T ← {courses in S that have no prerequisites in S};
    take all courses in T simultaneously;
    S ← S - T;
}
return semester;
```

30. First determine the worst-case running time of algorithm A as a θ function of n. Next determine the total running time of algorithm B as a θ function of n. Justify each answer.

A(n)	B(n)

if (n mod 2 == 1) return n;
else return A(n/2);
	for k ← 1 to n do
	print A(k);

Hint: Half of the n numbers are odd. Rest half are multiple of two, Rest half are multiples of 4 etc. Closed form needed for B(n): $\sum_{i=0}^{\infty} \frac{j}{2^j} \leq 2$

Answer:

Consider algorithm A. Suppose there are j calls to A before the recursion stops. This can only happen if $n/2^j \geq 1$, so we know $j \leq \lg n$, and hence $T_A(n) = \theta(\lg n)$. The worst case occurs when n is a power of 2.

Consider algorithm B. For n/2 values of k (the odd values), A is only called 1 time before the recursion stops. For n/4 values of k, A is called exactly 2 times. For $1 \leq j \leq \lg n$, there are $n/2^j$ values of k for which A is called exactly j times. Therefore $T_B(n) \leq \Sigma_{1 \leq j \leq \lg n}\, n/2^j * j \leq \Sigma_{1 \leq j \leq \infty}\, j*n/2^j = 2n = \theta(n)$.

31. **[6 pts]** You have *n* houses which you want to connect by fiber-optic cables. Your goal is to make sure that there is a route (possibly indirect, such as sending from house A to B then from B to C). For each pair of houses i; j, you know the cost c(i; j) for putting a cable between the two houses (cables allow two way traffic and c(i; j) = c(j; i)). You want a minimum cost collection of cables (minimize the sum of the costs of the cables selected) that allows routes between all the cities. How would you solve it efficiently? What is the running time?

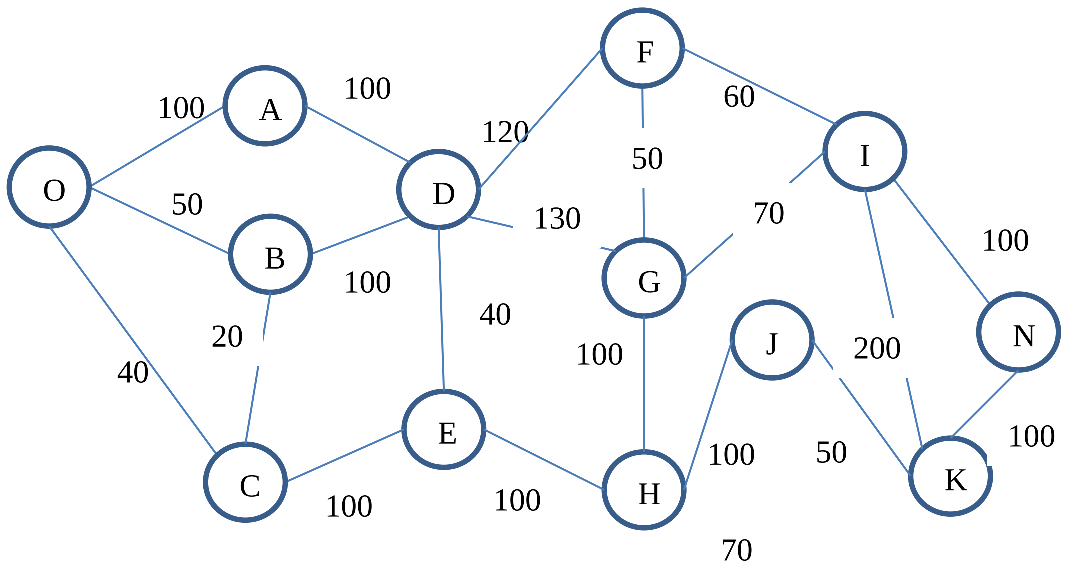
KEY:
This is a typical minimum spanning tree problem that can be solved by using Prim's or Kruskal's algorithm. Every house is a vertex and every possible fiber optic is an edge, the cost is the edge weight. The running time is O(|E|lg|V|).

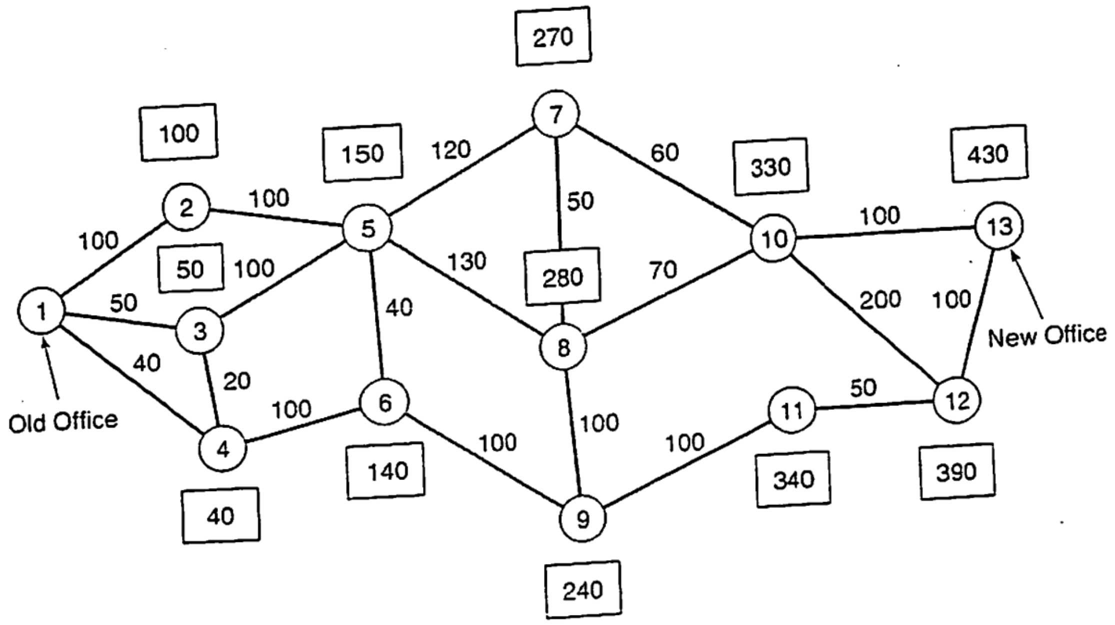
32. **[6pts]** You have a collection of 100 text files on your computer ranging in size from 100K bytes to several megabytes. The total size of all files is about 20 megabytes (so you can assume everything easily fits in main memory). Some files may be exact duplicates of each other. You want to identify any such duplicate files (so you can remove them). Describe how to find such files quickly (data structure, running time etc).

KEY:

Two files are identical if they have the same size.Let's say that we have n files. Obvious solutions is to compare each file to all other files O(n^2) or use sorting O(nlgn). A better solutions it to use a kind of hashing, calculate a hash value for every file size and remember that in a hash table. A hash table is an O(n) solution, which is faster than a sorted list for this task. Implement DuplicateFinder using a Hashtable that contains the pair <size,count>. Once populated, all files with count =1 will be removed:

33. **Hermes Moving Agency**
Hermes Moving has been hired to move the office furniture and equipment of Aphrodite Properties to their new headquarters (vertex N) from their old headquarters (vertex O). What route do you recommend? What is the algorithm that you use? Give all steps of your solutions in a table. The network of roads is shown next.

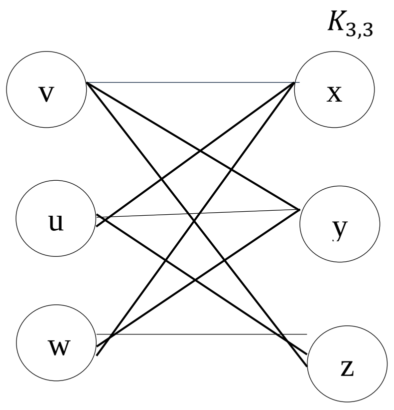




The shortest route is 1–3–5–7–10–13. The distance is 430 miles.

34. A matching in a graph is a set of disjoint edges, i.e. edges which do not share any vertices in common.
    a. **[8pts]** A complete bipartite graph $K_{3,3}$ =(V_1 +V_2 , E), is a bipartite graph such that for any two vertices v in V_1 and u in V_2, (v,u) is in E and $|V_1|=|V_2| = 3$. Give a simple example of a matching in a complete bipartite graph $K_{3,3}$ and highlight the edges that should be included in a maximum matching.
    [2pts] Can you think of an application for this problem?
    b. **[10pts]** Give an efficient algorithm to find a maximum matching in a tree. What is the running time of your algorithm

Solution:
Set V1={v,u,w} and V2={x,y,z} and one matching is (v,x), (u,y), (w,z)



Applications: Think of set V1 boys and set V2 girls and an edge means 'boys likes girl', then the matching is every girl with a boy.
Another one is V1 companies and V2 utilities

Algorithm:
```
M={}; // set that contains edges that form a matching
traverse the graph (use BFS or DFS),
while there are still edges,
        pick one edge, e.g., (v,x) and add it into M
        remove all edges incident to v and x, i.e. (v,y), (v,z), (x,u) and (x,w).
        (cannot be picked for M in the future,note: M contains disjoint edges)
Return set M
```

running time O(|V|+|E|)

35. Suppose that we have already computed a minimum-weight spanning tree M in a weighted undirected graph G. Now suppose we want decrease the weight of one of the edges of G not in M. In this case M may no longer be of minimum weight, and we may have to compute a new minimum-weight spanning tree. Describe a linear-time (O(n + m)) algorithm for this problem. Assume that the edges not in M are sorted by weight.

Solution:
Let (u, v) be the edge not in M whose weight decreased. Using DFS or BFS, find the unique simple path from u to v in M. Find an edge e of maximal weight on that path. If the weight of e is greater than that of (u, v), replace e in M with (u, v). Running time that of graph traversal O(|V|+|E|).