Anna Karlin, 426C Sieg

Practice Problems for Final
Not to be turned in!

1. Suppose that an algorithm A runs in worst-case time $f(n)$ and that algorithm B runs in worst-case time $g(n)$. For each of the following questions, answer either *yes, no,* or *can't tell.*
   - Is A faster than B for all n greater than some n_0 if $g(n) = \Omega(f(n)\log n)$?
   - Is A faster than B for all n greater than some n_0 if $g(n) = O(f(n)\log n)$?
   - Is A faster than B for all n greater than some n_0 if $g(n) = \Theta(f(n)\log n)$?

2. In the United States, coins are minted with denominations of 1,5,10,25, and 50 cents. Now consider a country whose coins are minted with denominations of $\{d_1, d_2, \ldots, d_k\}$ units.

   Give an efficient algorithm that correctly determines the minimum number of coins needed to make change of n units using the denominations $\{d_1, d_2, \ldots, d_k\}$. Analyze its running time.

3. A thief robbing a safe finds it filled with n types of items of varying size and value, but has only a small knapsack of capacity m to use to carry the goods. The knapsack problem is to find the combination of items which the thief should choose for his knapsack in order to maximize the total value of all the items he takes. Give a dynamic programming solution for the knapsack problem. What's the running time of your algorithm?

4. A company database consists of 10,000 sorted names, 40% of whom are known as good customers and who together account for 60% of the accesses to the database. There are two data structures options to consider for representing the database:
   - Put all the names in a single array and use binary search.
   - Put the good customers in one array and the rest in a second array. Only if we do not find the query name on a binary search of the first array do we do a binary search of the second array.

   Which option gives better performance?

5. In breadth-first and depth-first search, an undiscovered node is marked discovered when it is first encountered and marked completely explored when it has been completely searched. At any given moment, several nodes might be simultaneously in the discovered state. Describe a graph on n vertices and a particular starting vertex v such that during a breadth-first search starting from v $\Theta(n)$ nodes are simultaneously in the discovered state. Same question for depth-first search.

6. Suppose that G is a connected undirected graph. An edge e whose removal disconnects the graph is called a bridge. Must every bridge e be an edge in the depth-first search tree of G or can e be a back edge? Give a proof or a counterexample.

7. Is the path between a pair of vertices in a minimum spanning tree necessarily a shortest path between the two vertices in the full graph? Give a proof or a counterexample.

8. Dijsktra's shortest path algorithm maintains a set S of vertices whose final shortest path weights from the source have already been determined. For each node w in the graph there is a value $d[w]$ such that if w is in the set S, then $d[w]$ is the length of the shortest path from the source to w. What invariant holds for $d[w]$ for w not in the set S.

9. True or false: Suppose that E' is a subset of the edges of a graph such that there is a minimum spanning tree containing all edges in E' and let e be the minimum weight edge in the graph that is not contained in E'. Then $E' \cup e$ is also contained in some minimum spanning tree.

10. True or false: Let (u, v) be the minimum weight edge in the entire graph. Then (u, v) belongs to some minimum spanning tree.

11. True or false: A flow f in a flow network $G = (V, E)$ with source s and sink t is maximum if the value of the flow equals the capacity of some (s,t) cut in the graph.

12. True or false: The cardinality of a maximum matching in a bipartite graph G is the value of a maximum flow in its corresponding flow network G' (Define the corresponding flow network G'.)

13. Suppose you have a job that has been partitioned into a large number of smaller tasks, each with associated task time (the time it takes to complete the tasks). Suppose also that there are constraints on the tasks – for each pair of tasks, you are told whether it is essential that one task be performed before the other. Give an efficient algorithm for determining the minimum completion time for the entire job assuming an unlimited number of parallel processors (each able to complete a task the moment its precedence constraints have been satisfied).

14. Consider a telecommunications network with d source/destination pairs. For each pair s_i, t_i, there is demand that d_i units of data be transmitted between that source and destination. For every link (v, w) in the network, there is a bound $c(v, w)$ on the amount of data that can be transmitted across the link. The multicommodity flow problem is to find a flow for each of the different commodities (source/destination pairs) that satisfies all the demands without exceeding the total capacity of any edge. Set the problem up as a linear programming problem.

15. Show that a maximum flow in a network $G = (V, E)$ can always be found by a sequence of at most E augmenting paths. (Hint: Determine the paths *after* finding the maximum flow.)

16. Set up the problem of finding the length of the shortest path from node s to node t in a weighted undirected graph as a linear programming problem.

17. Perform one step in the simplex algorithm (on a linear programming problem of your choosing).

18. Is there an immediate polynomial time reduction from the traveling salesman problem on graphs to the Euclidean traveling salesman problem or vice versa?

19. What would be the significance of a program that could solve the traveling salesman problem in time proportional to 1.1^n.

20. It is an open question whether the decision problem "Is integer n a composite number?" can be computed in time polynomial in the size of the input. Why doesn't the following algorithm suffice to prove it is in P, since it runs in $O(n)$ time.

```
PrimalityTesting(n)
   composite := false;
   for i := 2 to n-1 do
     if (n mod i) = 0 then
         composite := true.
```

21. Suppose that two problems are known to be NP-complete. Does this imply that there is a polynomial-time reduction from one to the other?

22. The low-degree spanning tree problem is as follows. Given a graph G and an integer k, does G contain a spanning tree such that all vertices in the tree have degree at most k. Prove that the low degree spanning tree problem is NP-hard. (Hint: Hamiltonian path.)

23. Set up the knapsack problem as an integer linear program. Describe an approximation for the knapsack problem using randomized rounding.

24. Consider a branch and bound search to find the solution to the max-clique problem. How might you use quick-and-dirty bounds on the size of the maximum independent set in a graph to prune the search.

25. What are the essential differences between local search and search using simulated annealing?

26. True or false: The expected time for a successful search in a hash table with separate chaining is the same whether elements are inserted at the front or at the end of the list.

27. True or false: In double hashing all elements are stored in the hash table itself.

28. True or false: Consider a malicious adversary that chooses keys to be hashed in order to maximize the average search cost. If the adversary knows the hash function to be used,he can choose n keys to obtain an average search cost that is $\Omega(n)$.

29. What is the difference between hashing with separate chaining and hashing with open addressing?

30. Symmetric stream ciphers convert plaintext to ciphertext 1 bit at a time. Typically, the secret key is used as a seed for a pseudo-random generator (a deterministic algorithm that given a seed produces a stream of bits that hopefully appear to be random). The ciphertext is then produced by XORing the plaintext with the output of the pseudo-random generator. Describe how such a cipher might be vulnerable to known-plaintext attack or to ciphertext-only attack.

31. In the implementation of RSA, Bob chooses a random e between 1 and $\phi(n)$ such that $gcd(e, \phi(n)) = 1$. What is the probability that the first number e that Bob tries satisfies $gcd(e, \phi(n)) = 1$?

32. What is the "trapdoor" and what is the "one-way function" that forms the basis of the RSA public-key cryptosystem

33. Describe two situations in which you might use a one-way (cryptographic) hash function.

34. Describe two situations in which you might use a MAC.