Practice questions

Greedy algorithms

1. Prove by contradiction that for any cycle in a graph G whose edges have unique edge weights, the minimum spanning tree of G excludes the maximum-weight edge in that cycle.

2. Under a Huffman encoding of n symbols with frequencies $f_1, f_2, ..., f_n$, what is the longest a codeword could possibly be? Give an example of frequencies that could produce this case.

3. A server has n customers to serve. The required service time for each customer is known in advance: it is t_i minutes for customer i. So if, for example, the customers are served in increasing order of i, then the i-th customer will have to spend a total of $\sum_{k=1}^{i} t_i$ minutes to be completely done. We wish to minimize the total time spent by all the customers.

$$T = \sum_{i=1}^{n} (\text{time spent by customer } i)$$

   Give an efficient algorithm for computing the optimal order in which to process the customers.

Linear Programs

4. Duckwheat is produced in Kansas and Mexico and consumed in New York and California. Kansas produces 15 shnupells of duckwheat and Mexico 8. Meanwhile, New York consumes 10 shnupells and California 13. The transportation costs per shnupell are \$4 from Mexico to New York, \$1 from Mexico to California, \$2 from Kansas to New York, and \$3 and from Kansas to California. Write a linear program that decides the amounts of duckwheat (in shnupells and fractions of a shnupell) to be transported from each producer to each consumer, so as to minimize the overall transportation cost.

5. A film producer is seeking actors and investors for his new movie. There are n available actors; actor i charges s_i dollars. For funding, there are m available investors. Investor j will provide p_j dollars, but only on the condition that certain actors $Lj \subseteq \{1, 2, ..., n\}$ are included in the cast (all of these actors L_j must be chosen in order to receive funding from investor j).

   The producer's profit is the sum of the payments from investors minus the payments to actors. The goal is to maximize this profit.

   Question: Express this problem as an integer linear program in which the variables take on values $\{0, 1\}$.

6. What is the optimal solution to the following linear program:

$$\begin{aligned} \max \quad & -x + y \\ s.t \quad & 2x + 5y \geq 10 \\ & 2x - y \leq 20 \\ & y \leq 4 \end{aligned}$$

   A 5

   B 9

C ∞

D -8

E 4

Computational complexity

7. Consider the following problem, which is known to be NP-complete:

   Hamiltonian Cycle
   Input: Undirected graph $G = (V, E)$
   Question: Does G has a cycle that visits each vertex in V exactly once?

   Consider Hamiltonian Cycle restricted to graphs in which every vertex has at most 2 edges. Call this problem Hamiltonian Cycle-2.

   - Prove that Hamiltonian Cycle-2 is in NP.
   - What is wrong with the following proof of NP-completeness for Hamiltonian Cycle-2?
     We know that the Hamiltonian Cycle problem is NP-complete, so it is enough to present a reduction from Hamiltonian Cycle-2 to Hamiltonian Cycle. Given a graph G with vertices of degree at most 2, the reduction simply leaves the graph unchanged to be the input for Hamiltonian Cycle. The answer to both problems would clearly be identical. This proves the correctness of the reduction, thus Hamiltonian Cycle-2 is NP-complete.

8. (a) I have a problem X that I can solve by using 3SAT as a black box, with an additional polynomial amount of time. (That is, problem X poly-time reduces to 3SAT.) Are the following statements true or false/unknown?
      i. X is in NP.
      ii. X is NP-hard.
   (b) I have two problems, X and Y. 3SAT reduces to problem X and problem X reduces to problem Y. (These reductions are poly-time reductions.) Is the following statement true or false/unknown?
      i. Y is NP-hard.
   (c) Problem X and problem Y are both NP-complete. Is the following statement true or false/unknown?
      i. Problem X reduces to problem Y and vice versa.

9. For each of the statements in this section, respond:
   A True B False C Unknown

   (a) Problem A poly-time reduces to problem B. If problem A is solvable in a polynomial time then problem B is solvable in polynomial time.
   (b) Problem A poly-time reduces to problem B. If problem A is NP-complete then problem B is NP-complete.
   (c) Problem A poly-time reduces to problem B. If problem A is NP-hard then problem B is NP-hard.
   (d) If P $\neq$ NP, then the satisfiability problem (SAT) cannot be solved in polynomial time.
   (e) P $\subseteq$ NP.
   (f) NP-complete $\subseteq$ NP.
   (g) The intersection of NP-complete and P is empty.
   (h) If the satisfiability problem (SAT) poly-time reduces to problem Y, then Y is NP-hard.
   (i) NP-complete is a subset of NP-hard.

(j) If problem X is in NP and X poly-time reduces to the satisfiability problem (SAT), then X is NP-complete.

(k) No problems in NP can be solved in polynomial time.

(l) There are problems in NP that cannot be solved in polynomial time.

(m) If the satisfiability problem (SAT) can be solved in polynomial time, then all problems in NP can be solved in polynomial time.

For the last two questions, consider the following procedure:

```
procedure(X)
    Y[0] = X
    for i = 1 to k
        Y[i] = subroutine(Y[i-1])
```

Assume that `X` is valid input to `procedure`, `Y[i-1]` is valid input to `subroutine` for every `i`, and `subroutine` is a polynomial time algorithm.

(n) If `k` is a constant, then `procedure` is a polynomial time algorithm.

(o) If `k` is a polynomial function of the size of `X`, then `procedure` is a polynomial time algorithm.

10. We are feeling experimental and want to create a new dish. There are various ingredients we can choose from and we'd like to use as many of them as possible, but some ingredients don't go well with others. If there are n possible ingredients (numbered 1 to n), we write down an $n \times n$ binary matrix where the (i, j) entry is 1 if i and j can go together and 0 otherwise. Notice that this matrix is necessarily symmetric; and that the diagonal entries are always 0.

    We wish to solve the following problem:

    EXPERIMENTAL CUISINE :

    input: n, the number of ingredients to choose from; B, the $n \times n$ binary matrix that encodes which items go well together; and k the target number of ingredients.

    output: Can we come up with a dish with $\geq k$ ingredients that all go well with one another?

    Show that if EXPERIMENTAL CUISINE is NP-complete.