CS 231
Naomi Nishimura

Problems

Module 1

Finding a tour

Input: A complete graph with a weight on each edge

Output: A cycle that includes all the vertices (or a tour)

Tour Decision

Input: A complete graph with a weight on each edge, and a maximum budget B

Output: Yes or no, answering "Is there a tour of weight at most B?"

Tour Search

Input: A complete graph with a weight on each edge, and a maximum budget B

Output: A tour of weight at most B

Tour Counting

Input: A complete graph with a weight on each edge, and a maximum budget B

Output: The number of tours of weight at most B

Tour Enumeration

Input: A complete graph with a weight on each edge, and a maximum budget B

Output: The set of all tours of weight at most B

Tour Constructive Optimization

Input: A complete graph with a weight on each edge

Output: A tour of minimum weight

Tour Evaluation Optimization

Input: A complete graph with a weight on each edge

Output: The minimum weight of a tour

Finding distinct items

Input: A set of numbers

Output: The numbers in the set that appear exactly once

Travelling Salesperson Problem (TSP)

Input: A complete graph with a weight on each edge

Output: A tour of minimum weight

Pattern matching in a grid

Input: A pattern grid and a target grid

Output: Yes or no, answering "Does the pattern appear in the target?"

Module 2 Number 1 odd

Input: A sequence of numbers

Output: Yes or no, answering "Is 1 odd?"

First three odd

Input: A sequence of numbers of length at least three

Output: Yes or no, answering "Are the first three numbers all odd?"

All odd

Input: A nonempty sequence of numbers

Output: Yes or no, answering "Are all the numbers odd?"

Any odd

Input: A nonempty sequence of numbers

Output: An odd number in the sequence, or "None" if none exists

Maximum

Input: A nonempty sequence of numbers

Output: A number in the sequence with maximum value

Module 3

Scheduling activities

Input: A set of n activities, where activity i has start time s_i and finish time f_i

Output: A schedule that has the maximum number of nonoverlapping activities, that is, for each pair chosen $f_i \leq s_j$ or $f_j \leq s_i$

Making change

Input: A set of integer coin values $c_1, c_2, \ldots, c_k$ and an integer x

Output: Numbers of coins of each denomination such that the total number of coins is the smallest possible and the total value is x

Sorting

Input: A set of numbers

Output: The numbers in order from smallest to largest

Fractional knapsack

Input: A set of n types of objects, where object i has integer weight w_i and integer value v_i, and an integer weight bound W

Output: A list of fractions $0 \leq x_i \leq 1$ such that $\sum_{i=1}^{n} x_i \cdot w_i \leq W$ and $\sum_{i=1}^{n} x_i \cdot v_i$ is maximized

Single source cheapest paths

Input: A graph G with non-negative edge weights and a source vertex $s \in V(G)$

Output: The least-cost paths from s to each vertex in G, where the cost is the sum of the weights of edges in the path

Minimum spanning tree

Input: A connected graph G with non-negative edge weights

Output: A subset A of $E(G)$ that forms a tree from all of $V(G)$ and the sum of the weights of edges in A is as small as possible

k-Colouring

Input: A graph G and an integer k

Output: Yes or no, answering "Is G is k-colourable?"

3-Colouring

Input: A graph G

Output: Yes or no, answering "Is G is 3-colourable?"

Module 4

Questions

Input: A range of integers from a to b, $a \leq b$, and an integer k

Output: The only number in the range that is consistent with the answers to the k questions.

Maximum subtotal

Input: A sequence of n numbers a_1 through a_n

Output: $\sum_{k=i}^{j} a_k$ that is the largest possible

SELECTION

Input: A set of n numbers and a value j, $1 \leq j \leq n$.

Output: A number in the set that is the jth largest.

CLOSEST PAIR OF POINTS

Input: A collection of n points, where point $p_i = (x_i, y_i)$

Output: Points p_a and p_b in the collection, $a \neq b$, such that $\text{dist}(p_a, p_b)$ is minimized.

MATRIX MULTIPLICATION

Input: Two $n \times n$ matrices A and B

Output: The product of A and B

Module 5

MATRIX-CHAIN MULTIPLICATION

Input: A sequence (chain) of matrices $M_0, \ldots, M_{n-1}$, where M_i has dimension $d_i \times d_{i+1}$

Output: An order of multiplications that results in the smallest number of multiplications of pairs of values

LONGEST COMMON SUBSEQUENCE

Input: A string X of length m and a string Y of length n

Output: A string Z that is a subsequence of both X and Y and of maximum length

ALL-PAIRS CHEAPEST PATHS

Input: A graph G with non-negative edge weights

Output: The least-cost paths between each pair of vertices in G

KNAPSACK

Input: A set of n types of objects, where object i has integer weight w_i and integer value v_i, and an integer weight bound W

Output: A subset of objects with total weight at most W and the maximum total value

Module 6

ANY ODD

Input: A nonempty sequence of numbers

Output: Yes or no, answering "Does the sequence contain an odd number?"

Shortest Path Search

Input: A graph G and vertices u and v

Output: A shortest path from u to v

Shortest Path Decision

Input: A graph G and vertices u and v, and an integer k

Output: Yes or no, answering "Is there a path from u to v of length at most k?"

3-Colouring

Input: A graph G

Output: Yes or no, answering "Is G 3-colourable?"

Composite

Input: An integer N

Output: Yes or no, answering "Is N composite?"

Hamiltonian Cycle

Input: A graph G

Output: Yes or no, answering "Does G have a Hamiltonian cycle?"

Vertex Cover

Input: A graph G and an integer k

Output: Yes or no, answering "Does G have a vertex cover of size at most k?"

Independent Set

Input: A graph G and an integer k

Output: Yes or no, answering "Does G have an independent set of size at least k?"

Clique

Input: A graph G and an integer k

Output: Yes or no, answering "Does G have a clique of size at least k?"

Module 7

Integer knapsack

Input: A set of n types of objects, where object i has weight w_i and value v_i, and a weight bound W

Output: A list of integers $c_1, \ldots, c_n$ such that $\sum_{i=1}^{n} c_i \cdot w_i \leq W$ and $\sum_{i=1}^{n} c_i \cdot v_i$ is maximized

Partition

Input: A group of n objects, where object i has weight w_i

Output: A partition of the objects into groups A and B with minimum difference in the sum of the weights of each part

Module 8

Job Scheduling

Input: A set of jobs where job j takes time time(j)

Output: An assignment of jobs to m machines such that the makespan is minimized.

Bin Packing

Input: A group of n objects, where object i has size $s_i \leq 1$

Output: An assignment of objects to bins, where the sum of the sizes of the objects in a bin is at most 1, and the total number of bins is as small as possible