1. (DPV 3.11) Design a linear-time algorithm which, given an undirected graph G and a particular edge e, determines whether G has a cycle containing e.

* Find cycle in linear time

1. (DPV 3.16) Suppose a CS curriculum consists of n courses, all of them mandatory. The prerequisite graph G has a node for each course, and an edge from course v to course w if and only if v is a prerequisite for w. Find an algorithm that computes the minimum number of semesters necessary to complete the curriculum (assume that a student can take any number of courses in one semester). The running time of your algorithm should be linear.

* Shortest path on directed graph

1. (DPV 4.3) – Design and analyze an algorithm that takes as input an undirected graph G = (V, E) and determines whether G contains a simple cycle (that is, a cycle that doesn’t intersect itself) of length four. Its running time should be at most O(|V|^3).
2. (DPV 5.4) – Show that if an undirected graph with n vertices has k connected components, then it has at least n − k edges.
3. (DPV 5.7) – Show how to find the maximum spanning tree of a graph, that is, the spanning tree of largest total weight.
4. (DPV 5.13) – A long string consists of the four characters A, C, G, T; they appear with frequency 31%, 20%, 9%, and 40%, respectively. What is the Huffman encoding of these four characters?

The Huffman encoding will be:

100

/ \

T (40) 60

/ \

29 A (31)

/ \

G (9) C (20)

Now:

|  |  |  |
| --- | --- | --- |
| Letter | Frequency | Huffman Encoding |
| T | 40 | 0 |
| A | 31 | 11 |
| C | 20 | 101 |
| G | 9 | 100 |