

## Module 4: Basic Written

1. Suppose you are given a list of potential flights  $F = \{f_1, f_2, \dots, f_n\}$  that your airline wishes to serve. You can think of these as flight objects. Each flight has several fields including starting airport, ending airport, departure time, and arrival time. Develop an algorithm that, given flights  $F$  and a value  $k$ , determines whether it is possible to serve all of the flights in  $F$  with at most  $k$  planes. You must adhere to the following additional constraints:
  1. Plane maintenance takes  $m$  minutes. No plane can depart less than  $m$  minutes after arriving at an airport.
  2. A plane can fly between any two airports. For example, if you have two flights you want to serve in  $F$  (LA to BOS, NYC to DC) then we might choose to fly a plane from LA to BOS, then fly from BOS to NYC (even though that leg isn't in  $F$ ) so that this plane can then serve NYC to DC.
  3. You are given another parameter  $l$ . If you are splicing in an extra flight (see item above) then you must have at least  $m$  time for maintenance and  $l$  time for the other flight (yes, this is a pretty simple and unrealistic assumption, but is fine for our purposes).
  4. A plane does not need to end its day at the same airport from which it started. You can assume that all flights are during the day (between 6am and 12 midnight), and that there is an overnight flight at the end of the schedule back to the first departing airport before the next day begins.
2. The following problems are about reducing one problem to another in order to show one is NP-Hard. We define the k-DAG problem as follows: Given a directed graph  $G = (V, E)$  and value  $k$ , decide if a subset  $S \subseteq V$  of size  $k$  (and the edges attached to those vertices) can be removed such that the remaining graph has no cycles.

We will reduce k-Vertex-Cover (kVC) to k-DAG using the following construction. Recall that the inputs to kVC are an undirected graph  $G$  and value  $k$ . Given an instance of kVC,  $G = (V, E)$  and  $k$ , create a new directed graph  $G' = (V', E')$  for k-DAG where

- $V' = V$  (it has the same vertices as the kVC instance).
- For every edge  $(u, v) \in E$  in undirected graph  $G$  there will be two directed edges  $(u, v)$  and  $(v, u) \in E'$ .
- We pass the same  $k$  to k-DAG, i.e.,  $k' = k$ .

*Note: in a directed graph, two nodes  $u$  and  $v$  form a cycle if edges  $(u, v)$  and  $(v, u)$  exist.*

- **2.1** Explain why a given  $G$  will have a vertex cover of size  $k$  if k-DAG determines that  $G'$  has a subset of size  $k$  that, when removed, will remove all cycles.
- **2.2** We could ask you to prove the other direction of the relation described in the previous question, but were not asking that on this exam! But answer this: assuming we have also proved this relationship in the other direction, why has what we have done proved that k-DAG is NP-Hard?
- **2.3** Give a brief argument showing that k-DAG is also NP-Complete.