1. **We are given as input a set of n requests (e.g., for the use of an auditorium), with a known start time si and finish time ti for each request i. Assume that all start and finish times are distinct. Two requests conflict if they overlap in time — if one of them starts between the start and finish times of the other. Our goal is to select a maximum-cardinality subset of the given requests that contains no conflicts. (For example, given three requests consuming the intervals [0,3], [2,5], and [4,7], we want to return the first and third requests.) We aim to design a greedy algorithm for this problem with the following form: At each iteration we select a new request i, including it in the solution-so-far and deleting from future consideration all requests that conflict with i.**

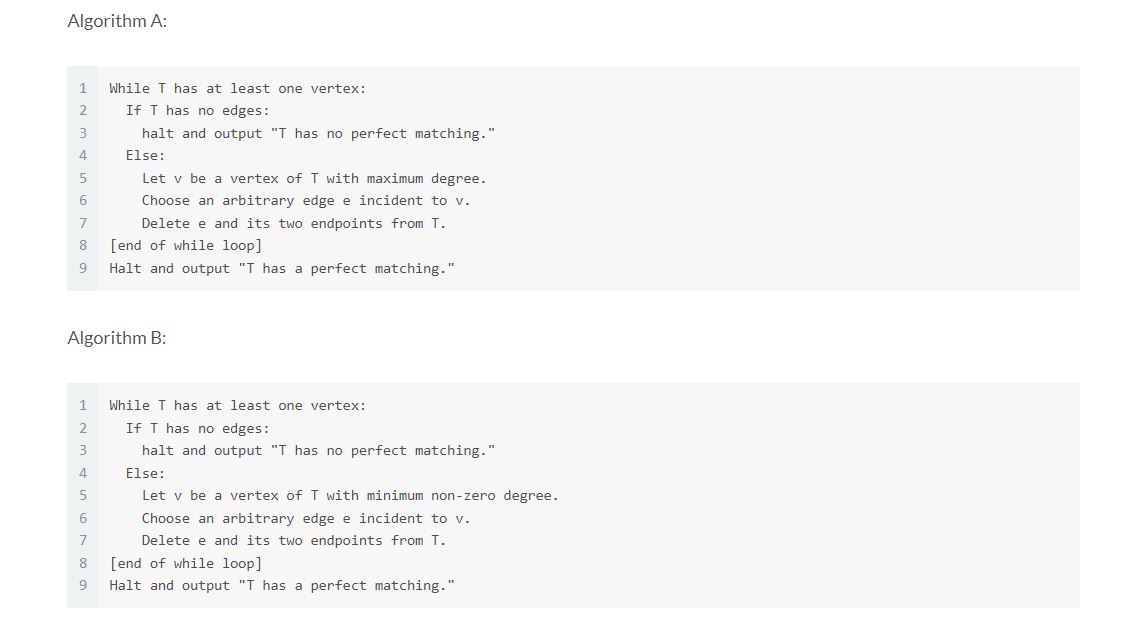
**Which of the following greedy rules is guaranteed to always compute an optimal solution?**

1. **We are given as input a set of n jobs, where job j has a processing time pj and a deadline dj. Recall the definition of completion times Cj from the video lectures. Given a schedule (i.e., an ordering of the jobs), we define the lateness lj of job j as the amount of time Cj−dj after its deadline that the job completes, or as 0 if Cj≤dj. Our goal is to minimize the maximum lateness, maxjlj.**

**Which of the following greedy rules produces an ordering that minimizes the maximum lateness? You can assume that all processing times and deadlines are distinct.**

1. **In this problem you are given as input a graph T=(V,E) that is a tree (that is, T is undirected, connected, and acyclic). A perfect matching of T is a subset F⊂⊂E of edges such that every vertex v∈V is the endpoint of exactly one edge of F. Equivalently, F matches each vertex of T with exactly one other vertex of T. For example, a path graph has a perfect matching if and only if it has an even number of vertices.**

**Consider the following two algorithms that attempt to decide whether or not a given tree has a perfect matching. The degree of a vertex in a graph is the number of edges incident to it. (The two algorithms differ only in the choice of v in line 5.)**



Solution : Algorithm B always correctly determines whether or not a given tree graph has a perfect matching; algorithm A does not. (Algorithm A can fail, for example, on a three-hop path. Correctness of algorithm B can be proved by induction on the number of vertices in T. Note that the tree property is used to argue that there must be a vertex with degree 1; if there is a perfect matching, it must include the edge incident to this vertex)

1. **Consider an undirected graph G=(V,E) where every edge e∈E has a given cost ce. Assume that all edge costs are positive and distinct. Let T be a minimum spanning tree of G and P a shortest path from the vertex s to the vertex t. Now suppose that the cost of every edge e of G is increased by 1 and becomes ce+1. Call this new graph G′. Which of the following is true about G′?**
2. **Suppose T is a minimum spanning tree of the connected graph G. Let H be a connected induced subgraph of G. (I.e., H is obtained from G by taking some subset S⊆⊆V of vertices, and taking all edges of E that have both endpoints in S. Also, assume H is connected.) Which of the following is true about the edges of T that lie in H? You can assume that edge costs are distinct, if you wish. [Choose the strongest true statement.]**