1. Apply the DFS-based topsort algorithm to linearize the following graph:
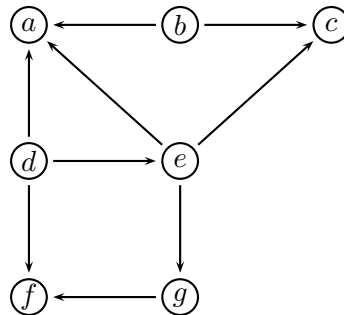
2. For what kind of array is the time complexity of insertion sort linear?

3. Trace how interpolation search proceeds when searching for 42 in an array containing (in index positions 0..21)

$$20, 20, 21, 23, 25, 26, 26, 27, 29, 29, 29, 30, 32, 33, 34, 36, 38, 40, 41, 43, 43, 45$$

4. Trace how QUICKSELECT finds the median of 39, 23, 12, 77, 48, 61, 55.

5. We can use QUICKSELECT to find the smallest element of an unsorted array. How does it compare to the more obvious way of solving the problem, namely scanning through the array and maintaining a variable min that holds the smallest element found so far?

6. **Just for fun ... a revision question**

   Consider the *subset-sum problem*: Given a set S of positive integers, and a positive integer t, find a subset $S' \subseteq S$ such that $\sum S' = t$, or determine that there is no such subset. Design an exhaustive-search algorithm to solve this problem. Assuming that addition is a constant-time operation, what is the complexity of your algorithm?