More graph algorithms

10/14/24

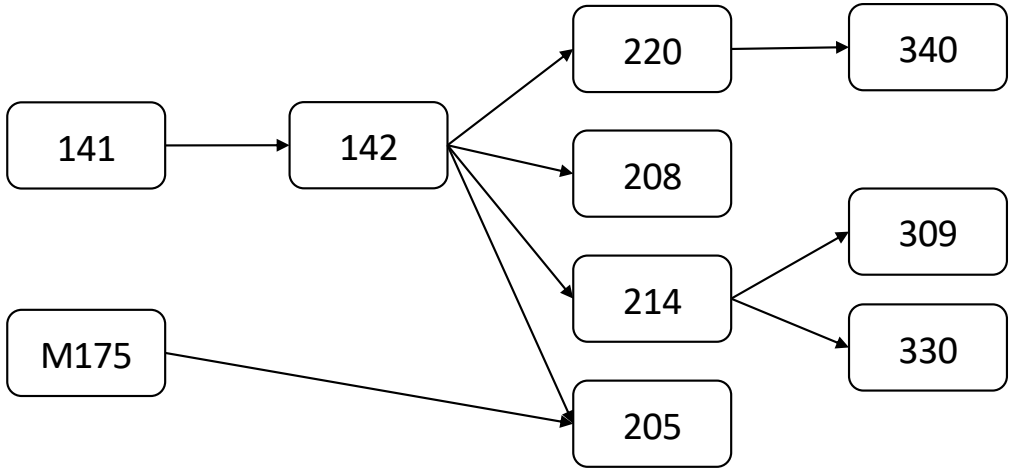
Administrivia

- HW 4 (Dynamic programming and Graph applications) due tomorrow night

- Expect:
  - Short HW 5 (due Friday) on the graph algorithms (including today) and network flow
  - Exam 2 is out Saturday; more multithreaded, dynamic programming, graph algorithms, network flow
    - Same general format (multi-day takehome with open written material and course webpage, closed everything else)
    - No class next Monday

Topological sorting

- Given a directed graph, create an order of the vertices so that all edges go to later vertices

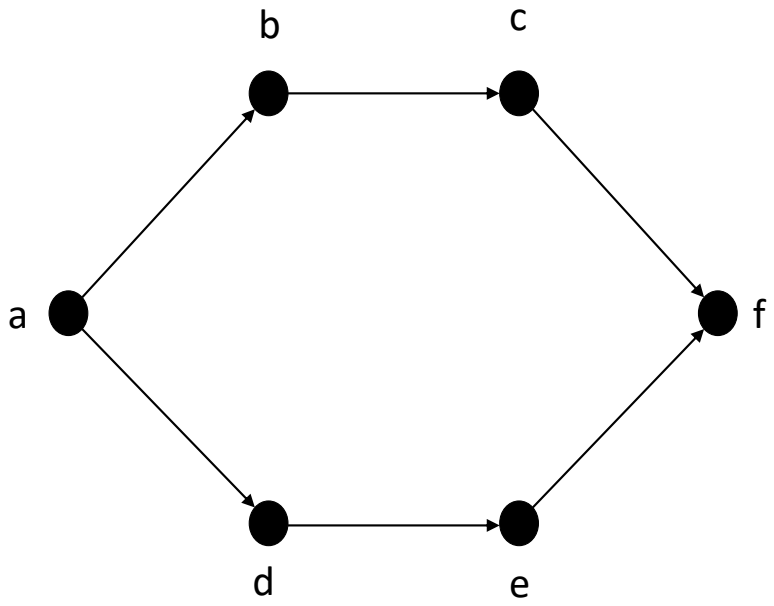


Possible order: 141, 142, M175, 208, 214, 309, 205, 220, 340, 330

How many topological orders does this graph have?

(Kleinberg & Tardos 3.1)

A. 2
B. 4
C. 6
D. 16
E. None of the above

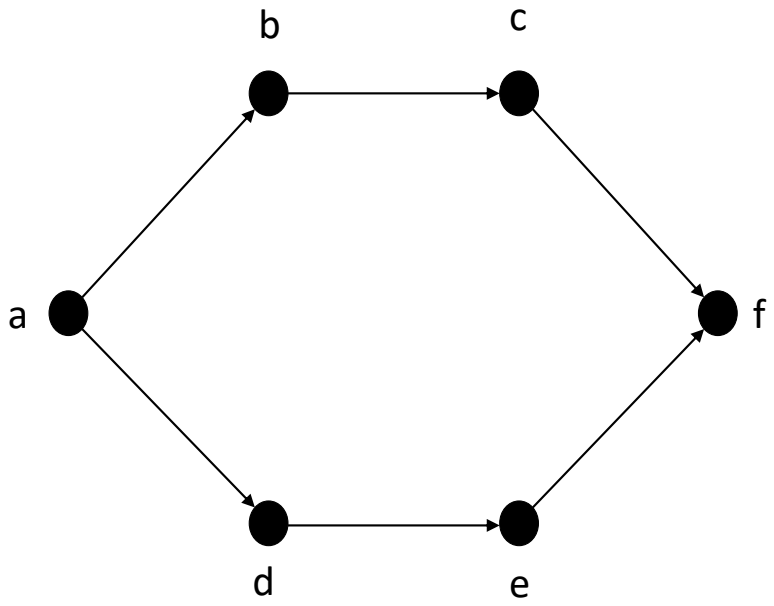


How many topological orders does this graph have?

(Kleinberg & Tardos 3.1)

A. 2
B. 4
C. 6
D. 16
E. None of the above

abcdef
abdcef
abdecf
adbcef
adbecf
adebcf



My favorite top sorting problem

(Kleinberg &Tardos 3.12)

You're helping a group of ethnographers analyze oral history data they've collected by interviewing members of a village. From the interviews, they've learned about n people $P_1, \ldots, P_n$. They've also learned facts of two forms about pairs of them:

- For some i and j, person P_i died before person P_j was born
- For some i and j, the life spans of person P_i and person P_j overlapped

Since they're based on memories, the researchers are not sure if all these facts are correct. Give an efficient algorithm to determine if the data they've collected is internally consistent in the sense that there could be a group of people for whom all the facts hold.

Finding a topological ordering

- One algorithm is based on DFS

- Another:
  - Initialize each vertex with its number of predecessors and a list of vertices without any predecessors
  - Repeatedly add vertices from the list to the output and decrement their successors' number of predecessors