COL 702 Lecture 23 Oct 27

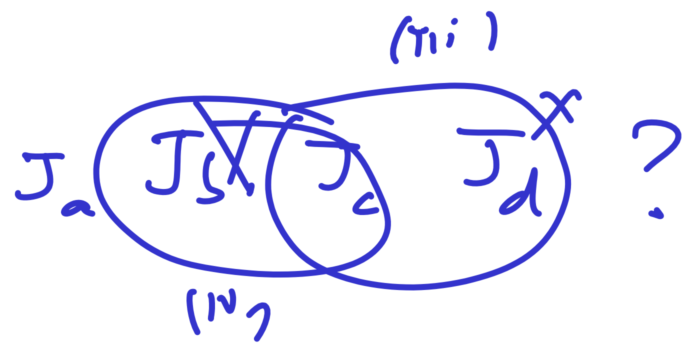
Given a set J_a J_b J_c J_d and some precedence contraints

(i) $J_a < J_b$: Job a must be done prior to Job b

(ii) $J_a < J_d$

(iii) $J_d < J_c$

(iv) $J_c < J_b$



Can all jobs be scheduled?

J_a J_d J_c J_b ✓

(i) $J_a < J_b$ (ii) $J_b < J_d$ (iii) $J_c < J_b$

(iv) $J_d < J_a$

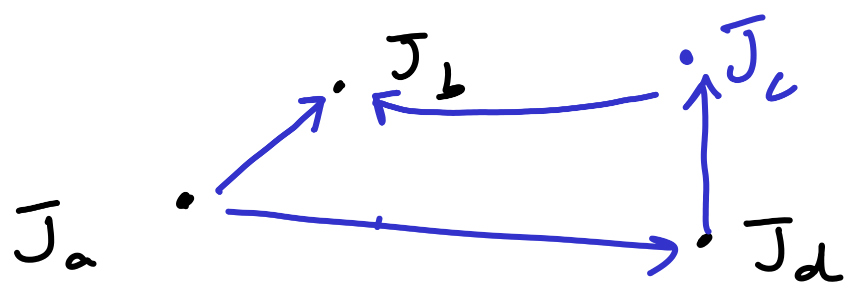
Not feasible since ($J_a < J_b < J_d$) there is cyclic precedence

<u>Observation</u> : There is no feasible schedule if there is a cyclic precedence between some subset of jobs.

If there is no cyclic precedence can we schedule?

Let us model it using a graph

V: set of Jobs E: (x, y) s.t $x < y$



$J_a \; J_d \; J_c \; J_b$

- Do a DFS. → always reveals if there is a cycle

If there is no cycle in a directed graph ⇒ Directed Acyclic Graph (DAG)

Observation: (i) There has to be a (source vertex) vertex with indegree 0

(ii) There must be a vertex with outdegree 0 (sink vertex)

A feasible schedule can be constructed by find a source, deleting it, and repeating this in the remaining graph. $O(|V|^2)$

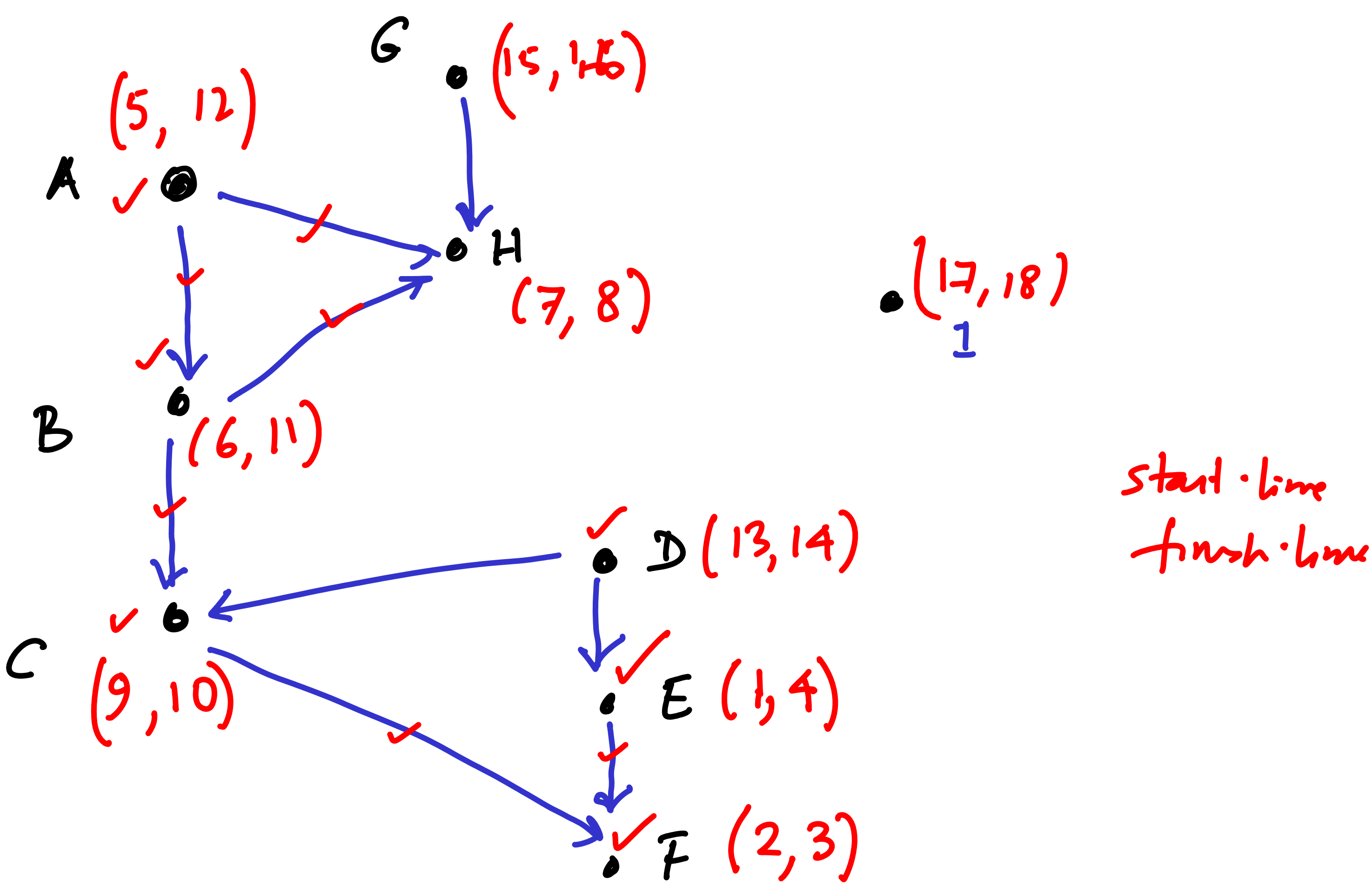
The precedence constraints define a "partial ordering"

Schedule: is a total ordering consistent with the partial ordering

Topological sort: Is an ordering of the vertices $v_1 \; v_2 \; v_3 \ldots v_n$ such that for any edge $v_i \to v_j$ $j > i$

By keeping track of indegrees of vertices, can we get a faster algorithm?

Which kind of data structure will yield a $O(|V|+|E|)$



Start time	E	F	A	B	H	C	D	G	I
Finish time	F	E	H	C	B	A	D	G	I

Decreasing finish · time may work!

Observation : Consider a pair of vertices v_i $(s_i\ f_i)$ and v_j $(s_j\ f_j)$

then if $v_i \rightsquigarrow v_j$ then

$$f_i \overset{?}{>} f_j$$

Case 1. $s_i < s_j$ then it will visit v_j during the DFS of v_i and back up from v_j before v_i

$$\Rightarrow f_i > f_j$$

Case 2 $s_j < s_i$ Since $v_j \not\rightsquigarrow v_i$ (DAG)

$$\underbrace{s_j < f_j} < \underbrace{s_i < f_i}$$

For any directed graph