

Incontro 2024-05-10: in aula I (DAGs)

Gli incontri avvengono sia in presenza che nella stanza Zoom:

<https://univr.zoom.us/j/85817283103>

E, quando disponibile, la loro registrazione è nel folder:

<https://univr.cloud.panopto.eu/Panopto/Pages/Sessions/List.aspx?folderID=c4df8587-8f34-4732-8e51-b12d0108ea08>

Prova symlink

<https://example.com>

<https://example.com>

[See example.com](#)

Directed Acyclic Graphs DAGs

Un *digrafo* è un grafo diretto $D = (V, A)$ dove ogni arco ha un verso. Un *topological sort* di un grafo diretto $D = (V, A)$ è un ordinamento dei suoi nodi $t : V \rightarrow \{1, \dots, n\}$ tale che per ogni arco $(u, v) \in A$ vale che $t(u) < t(v)$.

Il seguente è un problema di scheduling fondamentale:

1. dato un digrafo $D = (V, A)$, decidere se ammetta un topological sort e, in caso affermativo, produrne uno.

Siamo pervenuti insieme alla seguente buona congettura:

1. un DAG contiene un topological sort se e solo se non ha cicli

Che siamo riusciti a dimostrare (abbiamo visto/commentato più in generale come approcciare la dimostrazione di una buona congettura) pervenendo ad una buona caratterizzazione di cosa sia un DAG.

Questa comprensione ci ha concesso di risolvere su DAG problemi che su grafi generici sarebbero NP-completi.

L'idea è di sfruttare la struttura che caratterizza i DAG, ossia il loro topological sort. Esso si presta sia a produrre algoritmi greedy (per problemi particolarmente semplici) che algoritmi di programmazione dinamica.

Abbiamo osservato che l'intersezione di due DAGs $D_1 = (V, A_1)$ e $D_2 = (V, A_2)$, ossia $(V, A_1 \cap A_2)$, è di nuovo un DAG.

Ciò ci ha ripermesso di riscoprire Poldo come un problema di intersezione di due DAGs.