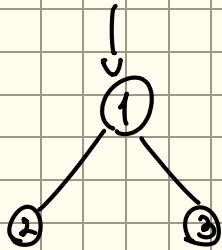
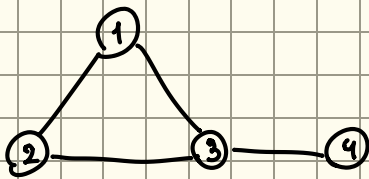
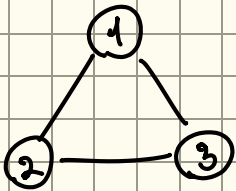


1. oct. 2025 curs 1

Subgraf indus
graf parțial



subgraf



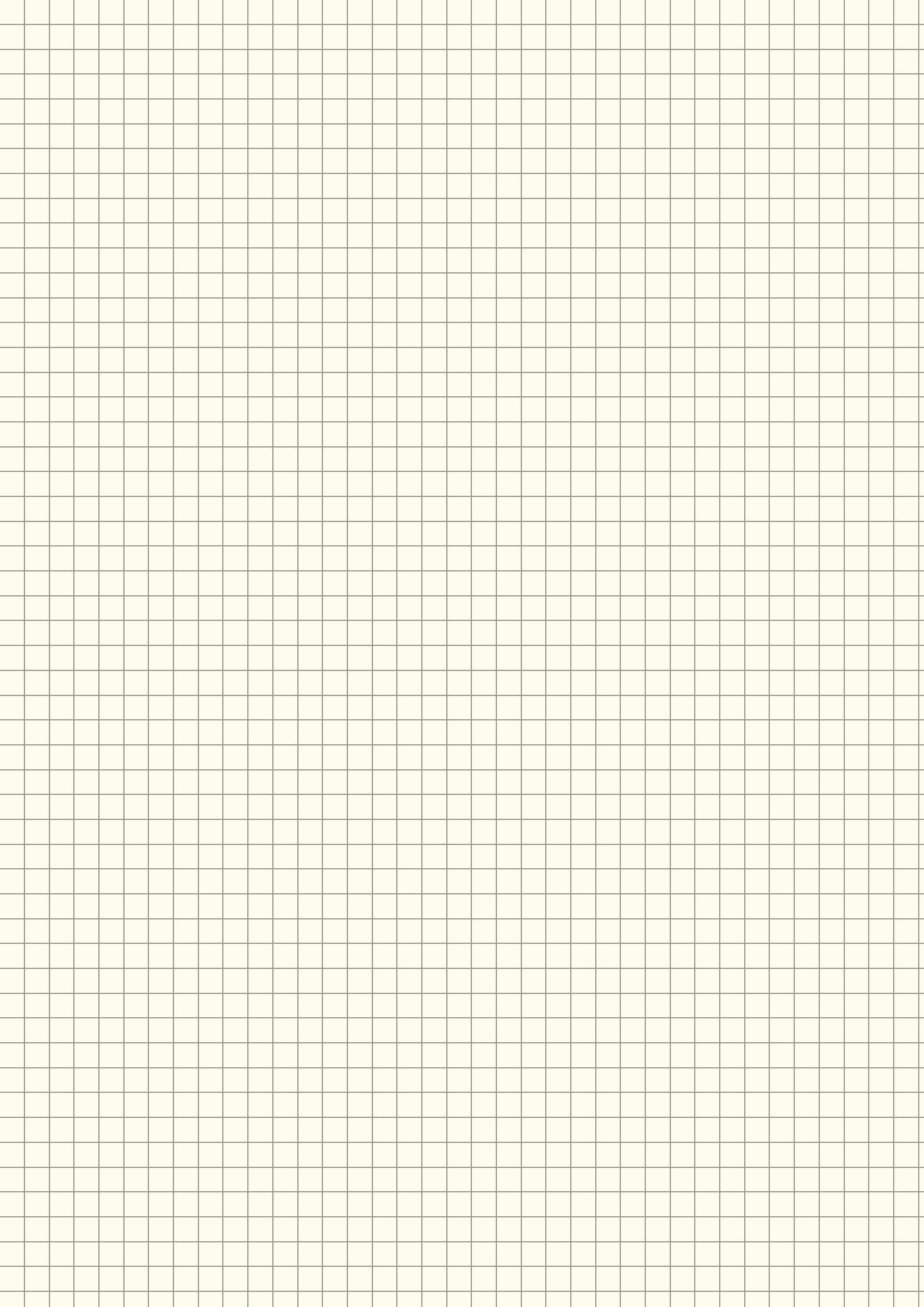
subgraf indus.

lanț (walk)

lanț simplu () \Rightarrow nu repetă muchii

lanț elementar (path) \Rightarrow nu repetă noduri.

K_n - graf complet cu n noduri.

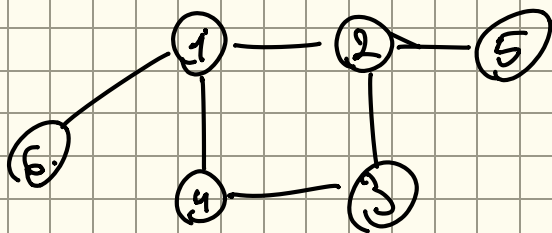


15 oct 2025 curs 3

Parcungeri

↗ Lățime
↘ adâncime

vf st: 1

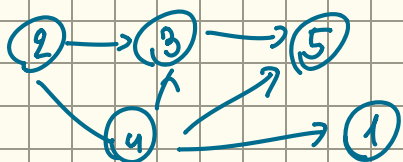


Sortare topologică

→ graf orientat

ST = 0 ond a vf cu prop că dacă $u \rightarrow v \in E$; u este înaintea lui v în ordonare

OBS 1) Dacă G admite 0 ST \Rightarrow nu are circuite



$S_1: 2, 4, 1, 5$

gr intern 0 rămâne cu gr in=0

Prop: G nu are circuite \Rightarrow poate fi sortat topologic

Dem: Alg (reducem pb. la un gr mai mic)

while $|V| > 0$
 alege v cu $d[v] = 0 \rightarrow$ exista mereu de G nu are circuite
 ad v la sortare
 $G \leftarrow G - v$
 ↓
 considerând dintr-unul max din G

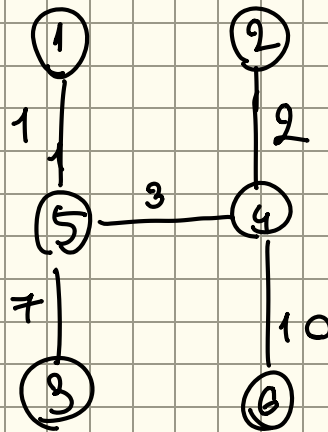
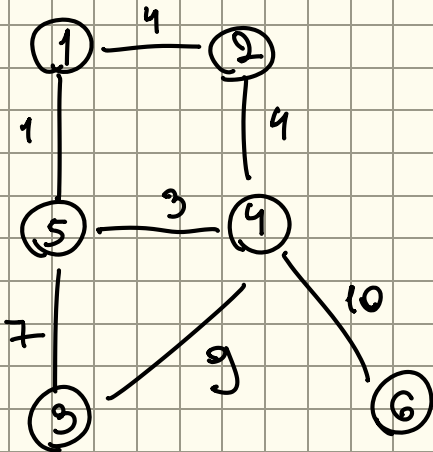
$0 \rightarrow 0 \rightarrow 0 \rightarrow 0 \rightarrow 0 \rightarrow 0 \rightarrow 0$
 ext inițială x are $d[x] = 0$
 (altfel putem extinde din sau se include un circuit)

22 oct 2025 curs 5

Arbori parțiali de cost minim $\rightarrow G$ conex ponderat
 \downarrow
 MST

$G = (V, E, W)$ - conex, pond $w : E \rightarrow \mathbb{R}$

Neponderat: arb part



$(u) \rightarrow (1,2)$

apcm nu este neapărat unic

Apcm în $G =$ un arb part T_{min}

în G cu

$w(T_{min}) = \min \{w(T) \mid T \text{ ap în } G\}$

Alg pt det apcm

Construcție apcm $\begin{cases} \text{prin adăugare} \\ \text{prin eliminare} \end{cases}$

1. Alg cu ad de muchii (build-up)

Schema $E(T) \leftarrow \emptyset$
pt $i = 1, n-1$

alegem o muchie $e \in E(T) \rightarrow$ criteriu ?
 $E(T) \leftarrow E(T) \cup \{e\}$

Kruskal:

$T \leftarrow (V, \emptyset)$

pt $i = 1, n-1$

alege $e \notin E(T)$ care nu
încide ciclu ca muchiile din $E(T)$
 $E(T) \leftarrow E(T) \cup \{e\}$

PRIM:

$T = (\{v\}, \emptyset)$

pt $i = 1, n-1$

alege $e = uv$ cu $u \in T$ și $v \notin T$

(o ext' vizitată cealaltă nu) (de cost minim).

$V(T) \leftarrow V(T) \cup \{v\}$

$E(T) \leftarrow E(T) \cup \{uv\}$

("gen" par)

Complex algorithm

cum mem. graful

Este utilă o procesare/ord inițială după cost,

Cum testăm că o muchie închide ciclu ca muchiile deja selectate.

(V1) Parcurgere în T pornind din u și verificăm dacă v este vizitat.

(V2) Verif dacă u și v sunt în comp conexe din T dif \rightarrow mem mulțimea vf din fiecare componentă.

Fiecare mult - are asociat un reprezentant/culoare/didacta

Struct pt. op cu mulți

Operații \rightarrow care este rep mult lui $u \rightarrow \text{find}(u)$
 \rightarrow reuniunea a două mulțimi $\rightarrow \text{union}(u, v)$

1) Sontesa (E)

2) pt. fiecare $u \in V$ $\text{Init}(u)$ $\{u\}$

3) pt. $u, v \in E$ (în ord cresc)

dacă $\text{find}(u) \neq \text{find}(v)$

$\text{ad}(u, v)$ la T

$\text{union}(u, v)$

dacă $|E(T)| = n-1$ stop

① Mem cc(mult și lor) cu vector reprezent.

$r[u] = \text{rep}(\text{culoarea mul } u)$

$\text{init}(u) \quad r[u] = u$

5 noiembrie curs 6

APCM

