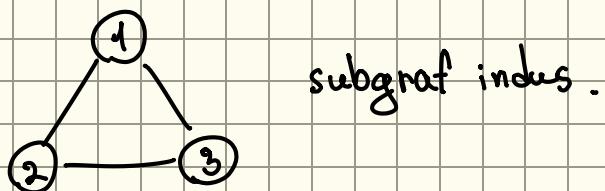
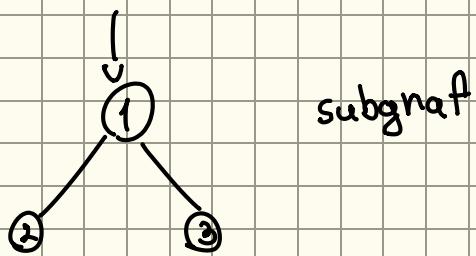
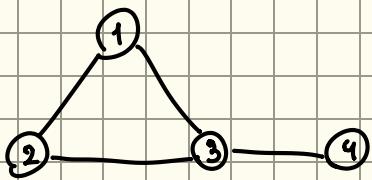


Subgraf induș
graf parțial

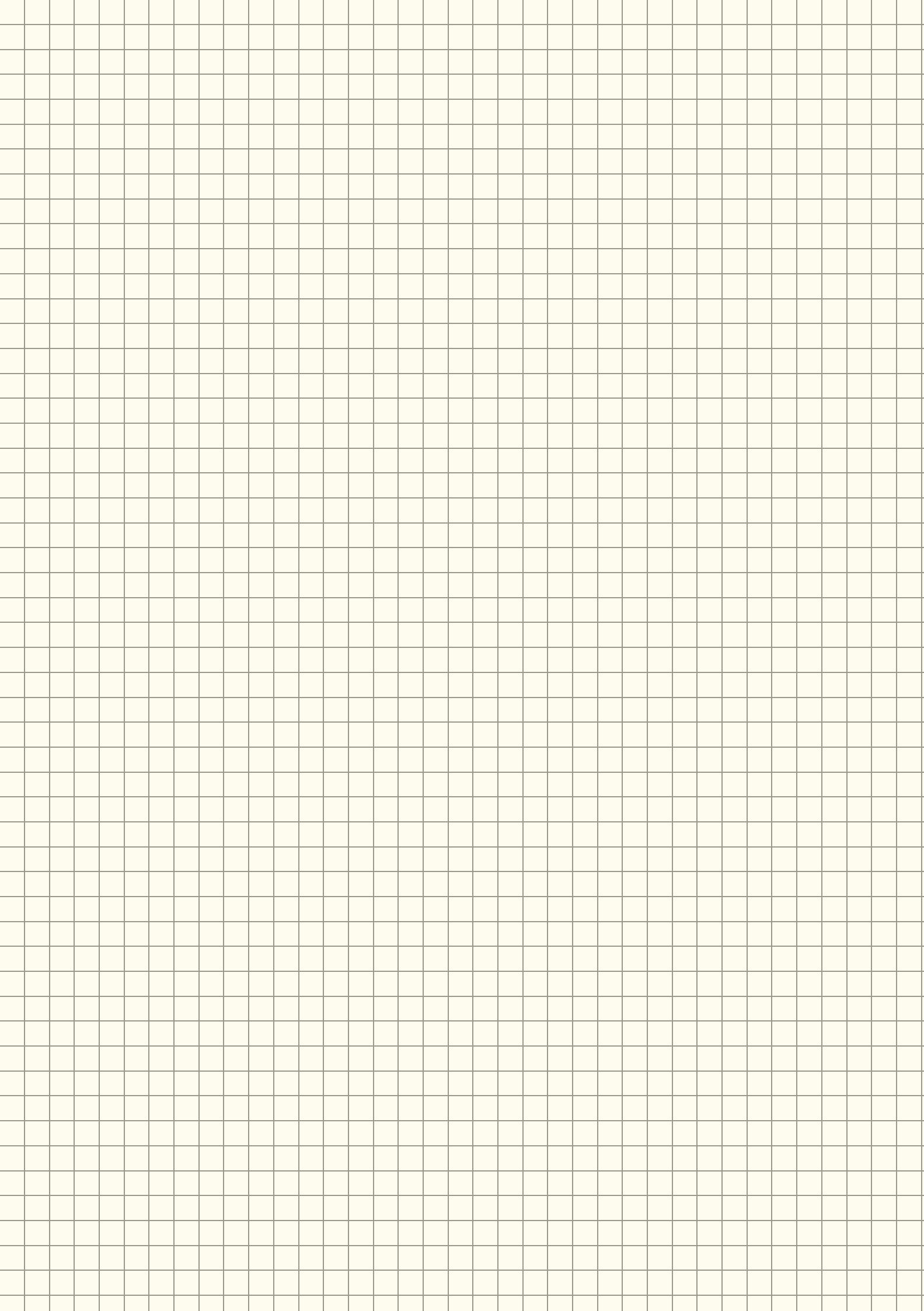


lant (walk)

lant simplu () \Rightarrow nu repetă muchii

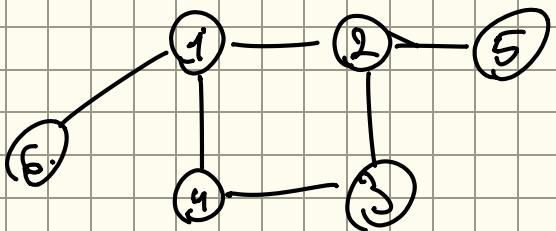
lant elementar (path) \Rightarrow nu repetă noduri.

K_n -graf complet cu n noduri.



Parcurgeri

→ Lățime vfst: 1
 → adâncime

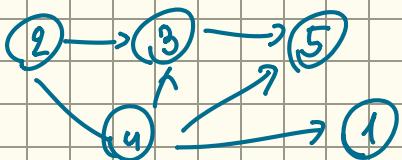


Sortare topologică

→ graf orientat

$ST = \emptyset$ și daca $v \in ST$ și u este vecinul lui v în ordonare

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



$S_1: 2, 4, 1, 5$
 numărătoare cu
 înainte înainte

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

Dem: Algl reducem pb. la un gr mai mic)

while $|V| > 0$
 aleg v cu $d[v] = 0 \rightarrow$ există mereu de G nu are circuite

ad v la sortare
 $G \leftarrow G - v$

Considerând din nou max din G

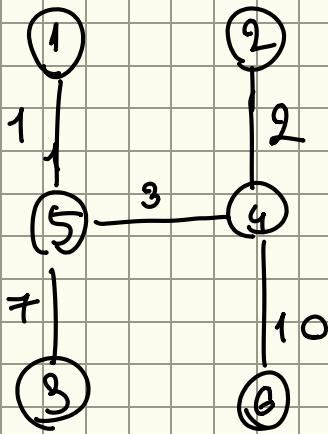
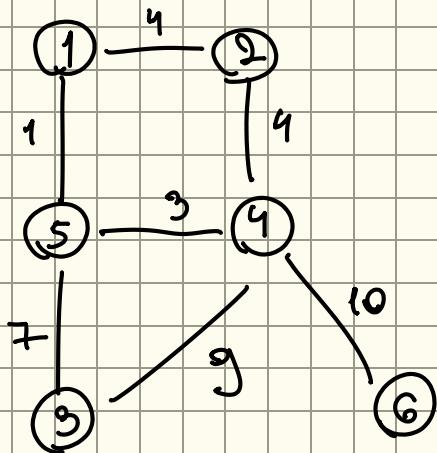
$0 \rightarrow 0 \rightarrow 0 \rightarrow 0 \rightarrow 0 \rightarrow 0 \rightarrow 0$
 ext initială \times are $d[x] = 0$
 (atfel patem extinde dn sau se include un circuit)

22 oct 2025 curs 5

Arborei parțiali de cost minim \rightarrow G conex ponderat
 \downarrow
 MST

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

Neponderat: arb part



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

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

prin adăugare
prin eliminare

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

Schemă $E(T) \leftarrow \emptyset$

pt $i = 1, n-1$

alegem o muchie $e \in E(T)$

criteriu ?

Kruskal: $T \leftarrow (V, \emptyset)$

pt $i = 1, n-1$

alegem $e \in E(T)$ care nu

închide ciclu ca muchiile din $E(T)$

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

PRIM: $T = (V, \emptyset)$

pt $i = 1, n-1$

alegem $e = u \rightarrow v$ cu $u \neq v$

în T și oalătă nu

(u este vizitată oalătă nu) (de cost minim).

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

$E(T) \leftarrow E(T) \cup \{u \rightarrow v\}$

(gen "parc")

Complex algoritm

cum mem. grafal

Este utilă o procesare (ord inițială după cost).

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

V1 Parcurgerea în T pornind din u și verificăm dacă v este vizitat.

V2 Verificăm dacă $u \rightarrow v$ sunt în compconexe din T dif \rightarrow mem multimea U_f din fiecare componentă.

Fiecare mult - are asociat un reprezentant/culoare/eticheta

Struct pt. de cu multi

Operatii \rightarrow care este rep mult lui $u \rightarrow \text{find}(u)$

\rightarrow reunirea a două multiimi $\rightarrow \text{union}(u, v)$

1) Sortează (E)

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

3) pt. $u \neq v$ (in ord cresce)
dacă $\text{Find}(u) \neq \text{Find}(v)$
mult $\rightarrow u$

$\text{ad}(u, v) \rightarrow T$
 $\text{UNION}(u, v)$
dacă $|E(T)| = n-1$ stop

V1 Mem $\alpha(\text{mult} \vee \text{lon})$ cu vector reprezent.

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

$\text{init}(u) \pi(u) = u$

5 noiembrie curs 6

APCM

