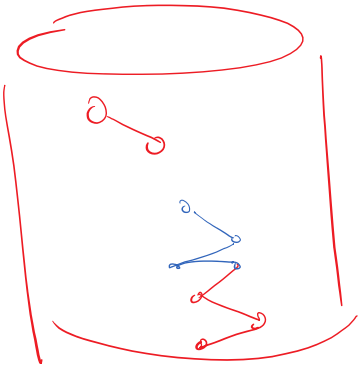
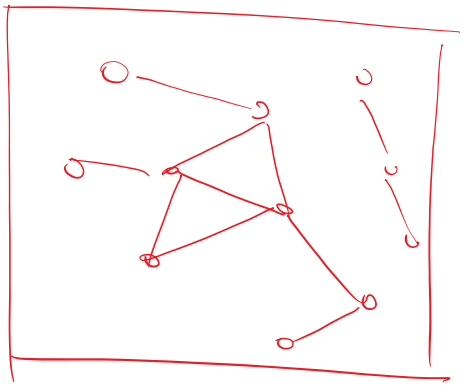
graph pattern mining



Database of graphs

vs.



One large graph (social graph)

labeled graphs

$G = (V, E)$

$L(u)$ = label of vertex u

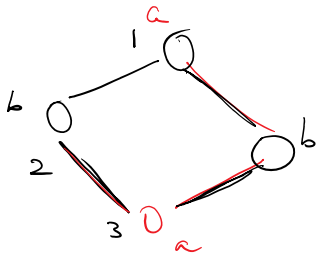
$L(u,v)$ = label of an edge (u, v)

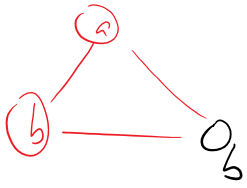
Task: find commonly occurring subgraphs

frequent subgraphs — pattern

$freq(P) \equiv Sup(P) =$ { # of distinct graphs G_i that contain P ; # of occurrences of P over all $G_i \in D$ }

$freq(P)$: frequency; $Sup(P)$: support; D: graph database





P:

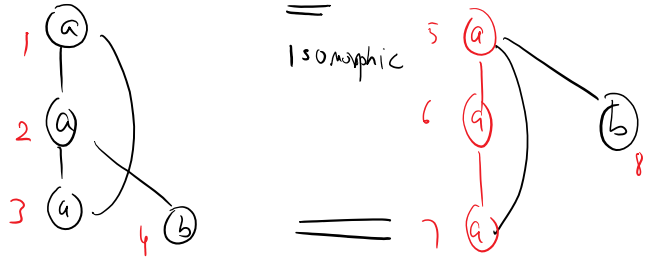
$Sup(P) = 2$ (distinct graphs)

$Sup(P) = 6$ (total # of occurrences)

Mining Task: given a minimum support threshold θ find <u>all</u>

frequent subgraphs, i.e. $Sup(P) \geq \theta$

potentially exponential space



1	2	3	4	$\leftarrow x \in G_1$
6	5	7	8	$\phi(x) \in G_2$
7	5	6	8	

Graph G_1 is <u>isomorphic</u> to G_2 if

$\rightarrow \phi$: 1-1 correspondence between G_1 & G_2 both into and onto

such that

$x \in G_1$

$\phi(x) \in G_2$

① Structure has to be preserved

$$(x,y) \in G_1 \iff (\phi(x), \phi(y)) \in G_2$$

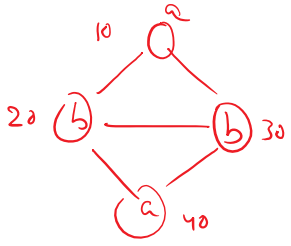
② labels have to be preserved

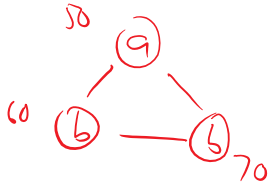
$$L(x) = L(\phi(x))$$

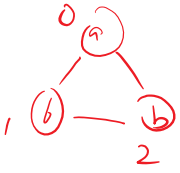
$$L(x,y) = L(\phi(x), \phi(y))$$

Graph isomorphism problem

is it in P?







	a	b	b
	0	1	2
G_1	10	20	30
	10	30	20
	40	20	30
	40	30	20
G_2	50	60	70
	50	70	60

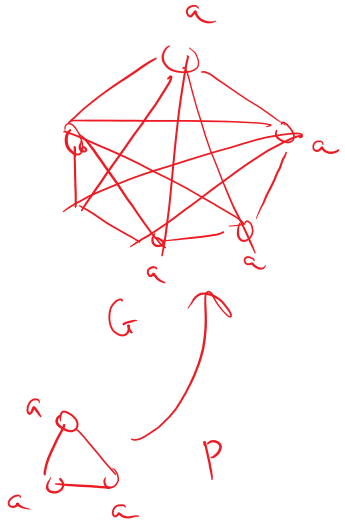
Subgraph isomorphism

NP-hard problem

ϕ : subgraph isomorphism

embedding : $(\phi(x_1), \phi(x_2), \ldots, \phi(x_n))$ ← G_i

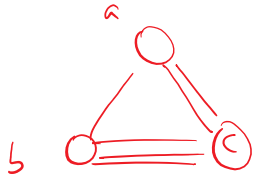
$x_1\ x_2 \ldots x_n$ ← P



① Systematic search

→ extend a pattern by one extra edge

→ 1) pattern extension step

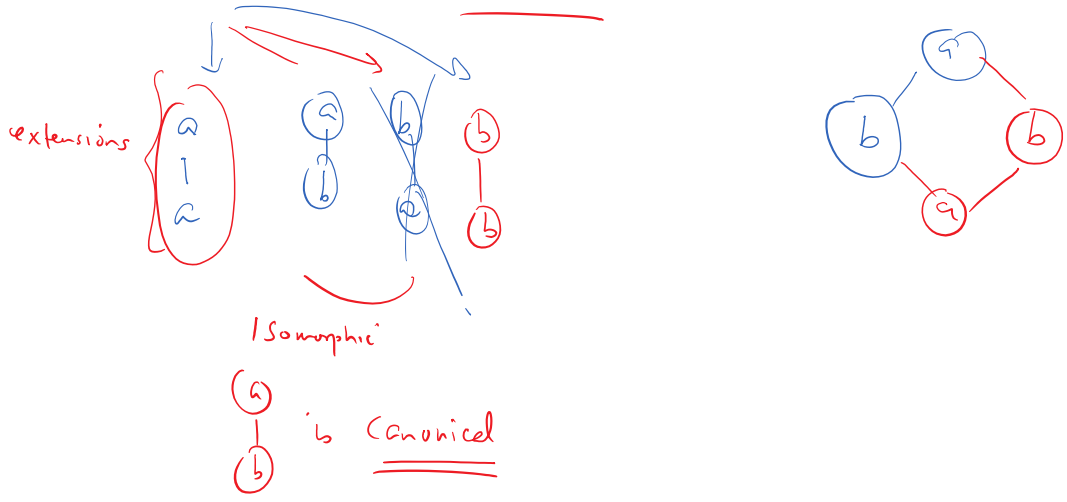


→ check for "duplicates": list/count only the distinct subgraphs/patterns
2) (graph isomorphism)

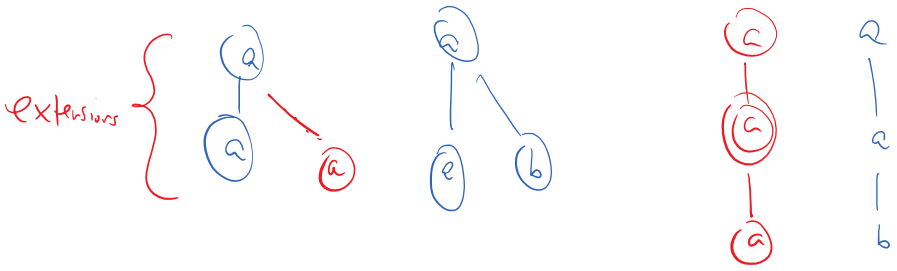
(2) Collect the frequency for each pattern P
→ 3) subgraph isomorphism step

Step 1: Candidate generation / Pattern extension

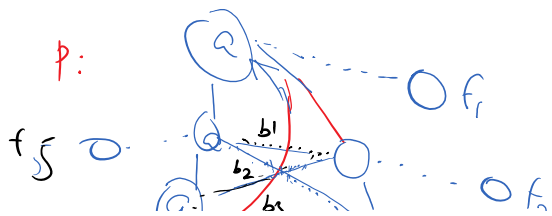
P: ∅ $L \in \{a, b\}$

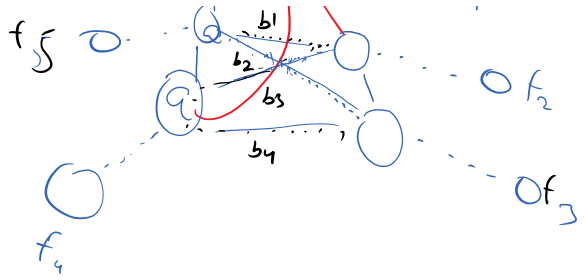


P:



P:





$S \times |L|$

forward extensions

$f_1 \ldots f_5$

add new nodes.

backward extensions

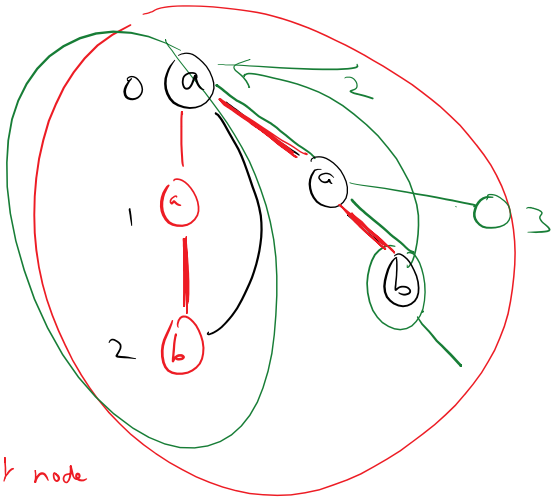
$b_1 \ldots b_4$

add cycles

Every pattern will be represented by its

Canonical Code

DFS code → DFS tree



extend

① go depth-first do a forward extension from the current node

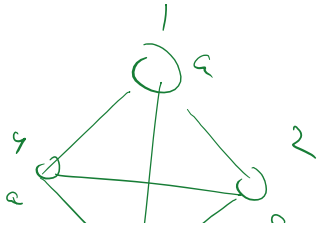
② add backwards edges closer to root first from the last vertex.

③ start branching bottom to top

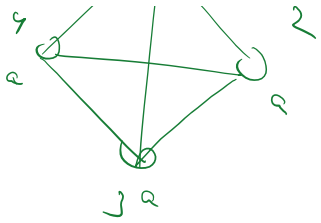
Only to right most path

$S = \{ G_1, G_2, \ldots, G_n \}$

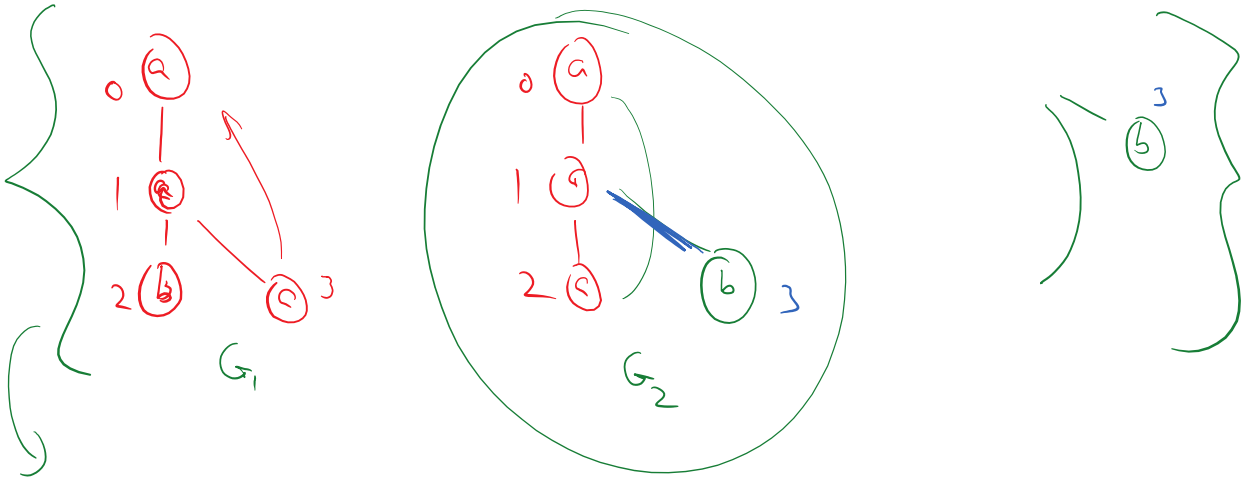
isomorphic graphs



Isomorphic graphs



min DFS code (S) → a unique canonical representative in terms of the DFS-tree



DFS code (G1)

(0, 1, a, a, –)
(1, 2, a, b, –)
(1, 3, a, c, –)
(3, 0, a, c, –)

X

0 1 a a –
1 2 a a –
2 0 a a –
1 3 a b –

DFS code (G2)

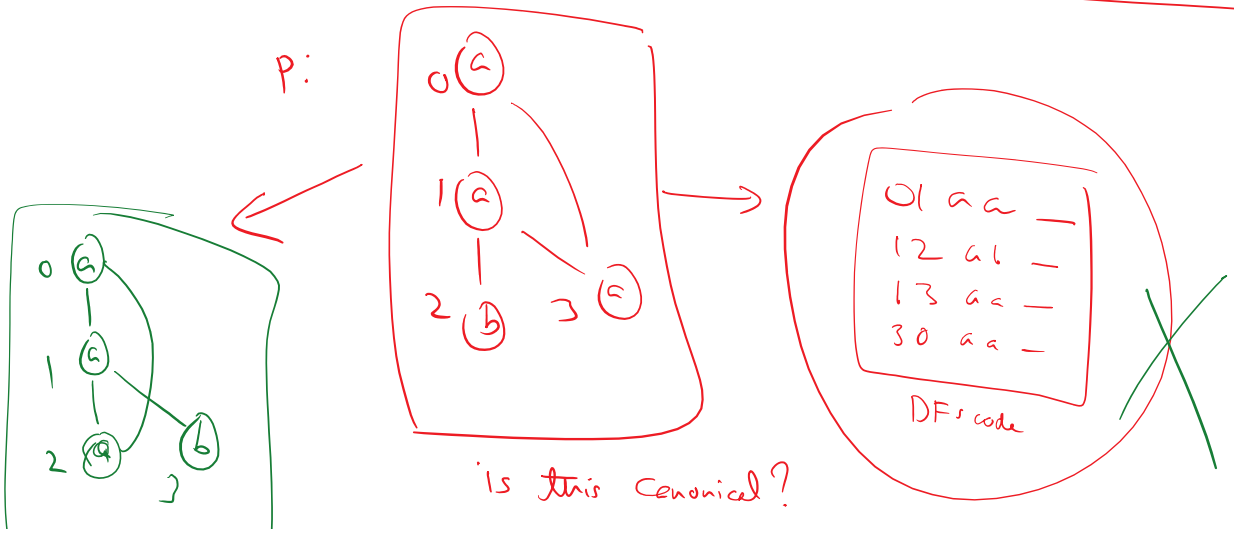
✓

0 1 a a –
1 2 a a –
2 0 a c –
0 3 a b –

DFS code (G3)

X

P:



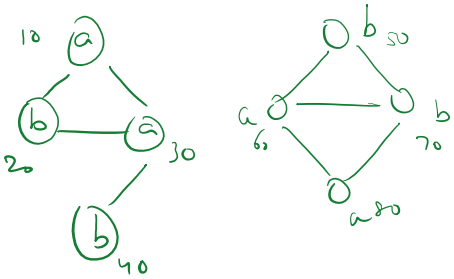
is this canonical?

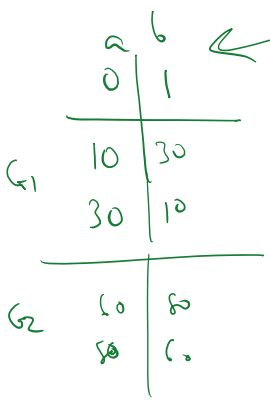
DFS code

0 1 a a –
1 2 a b –
1 3 a c –
3 0 a a –

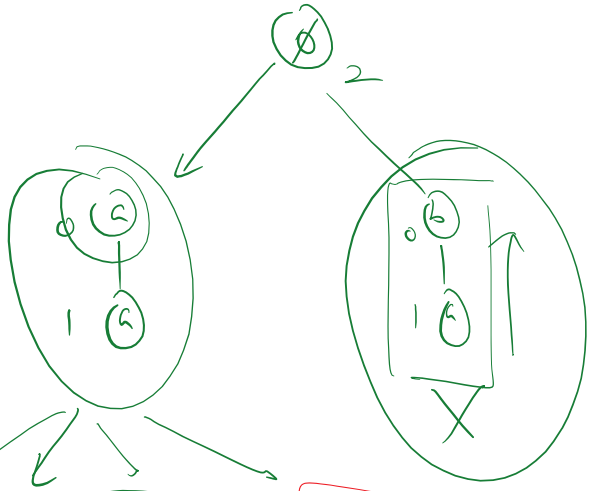
Is this canonical?

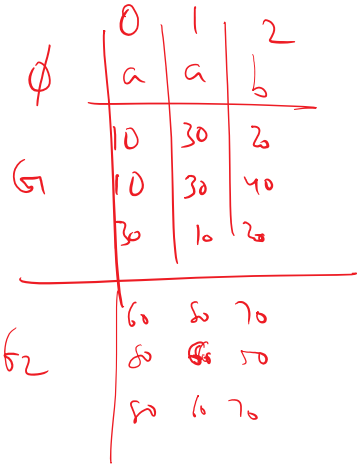
→ is this a duplicate?

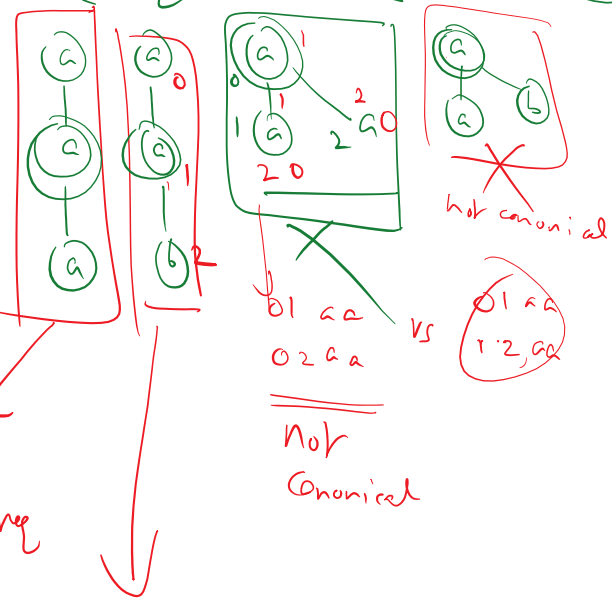




minsup = 2







not canonical

01 aa vs 01 aa, 12 aa

not canonical