Esercizio 35.1.4 CORMEN

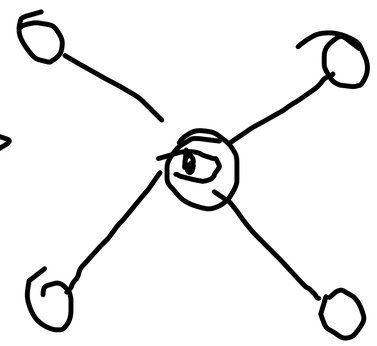
Dato un albero di n vertici connesso e rappresentato come una lista di adiacenza. Trova un algoritmo in grado di fornire un vertex cover ottimo in $O(n)$

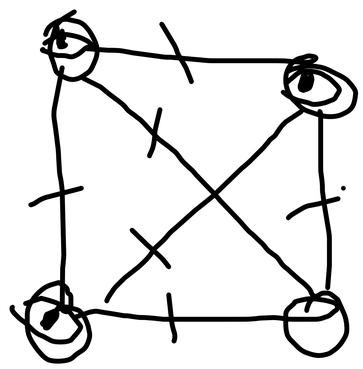
Il problema di solito non è risolvibile polinomialmente se non in maniera approx.

Però ci sono dei casi facili

Stella

1



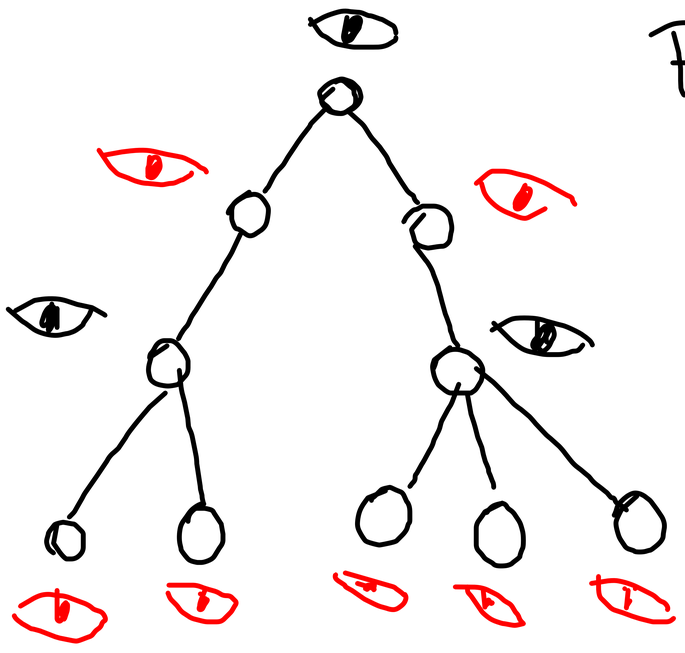


Clique

n-1

Esercizio 35.1.4 CORMEN

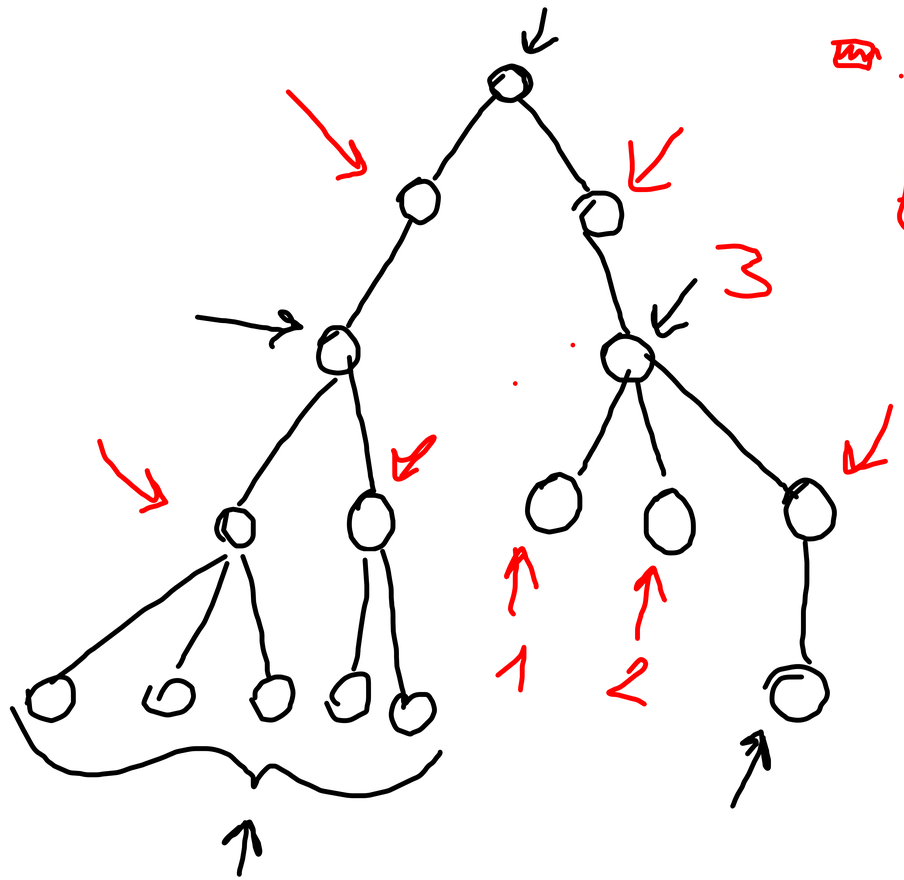
Dato un albero di n vertici connesso e rappresentato come una lista di adiacenza. Trova un algoritmo in grado di fornire un vertex cover ottimo in $O(n)$



Proviamo a colorare un livello sì ed un livello no.

Non è detto che la tecnica un livello sì e uno no ci porti ad una sol. ottima

↳ Sembra che: Ø un VERTEX-COVER ottimo che contiene delle foglie



■ : Nwhite = 9

▭ : Nred = 7

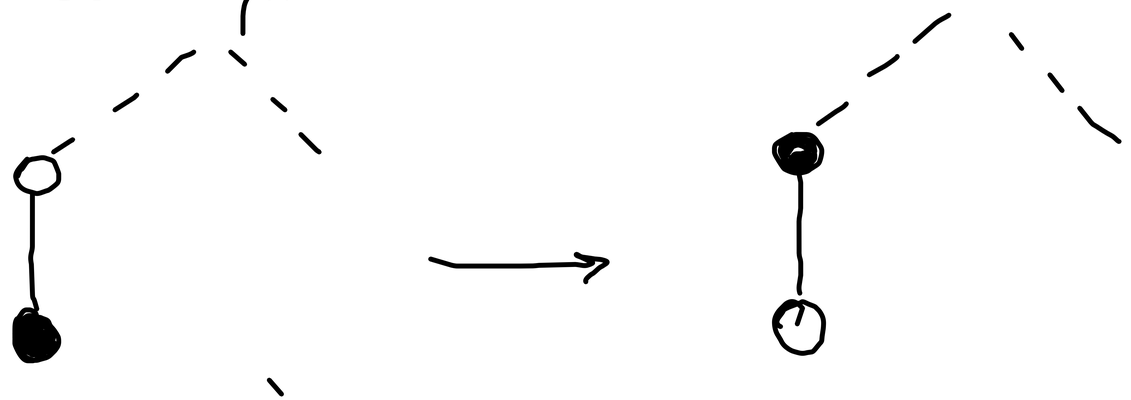
Esiste uno SCAMBIO che migliora la nostra soluzione:

Far diventare 3 rosso invece di 1 e 2.

Scegliere un genitore delle foglie sembra una strada promettente da percorrere.

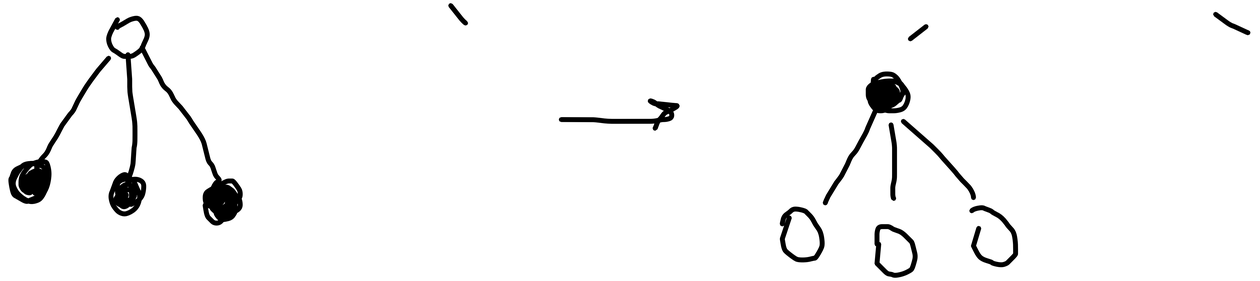
In generale, inserendo un genitore all'interno della soluzione al posto dei suoi figli (che sono foglie) otteniamo sempre una sol. NON PEGGIORE di quella corrente

NON PEGGIORE



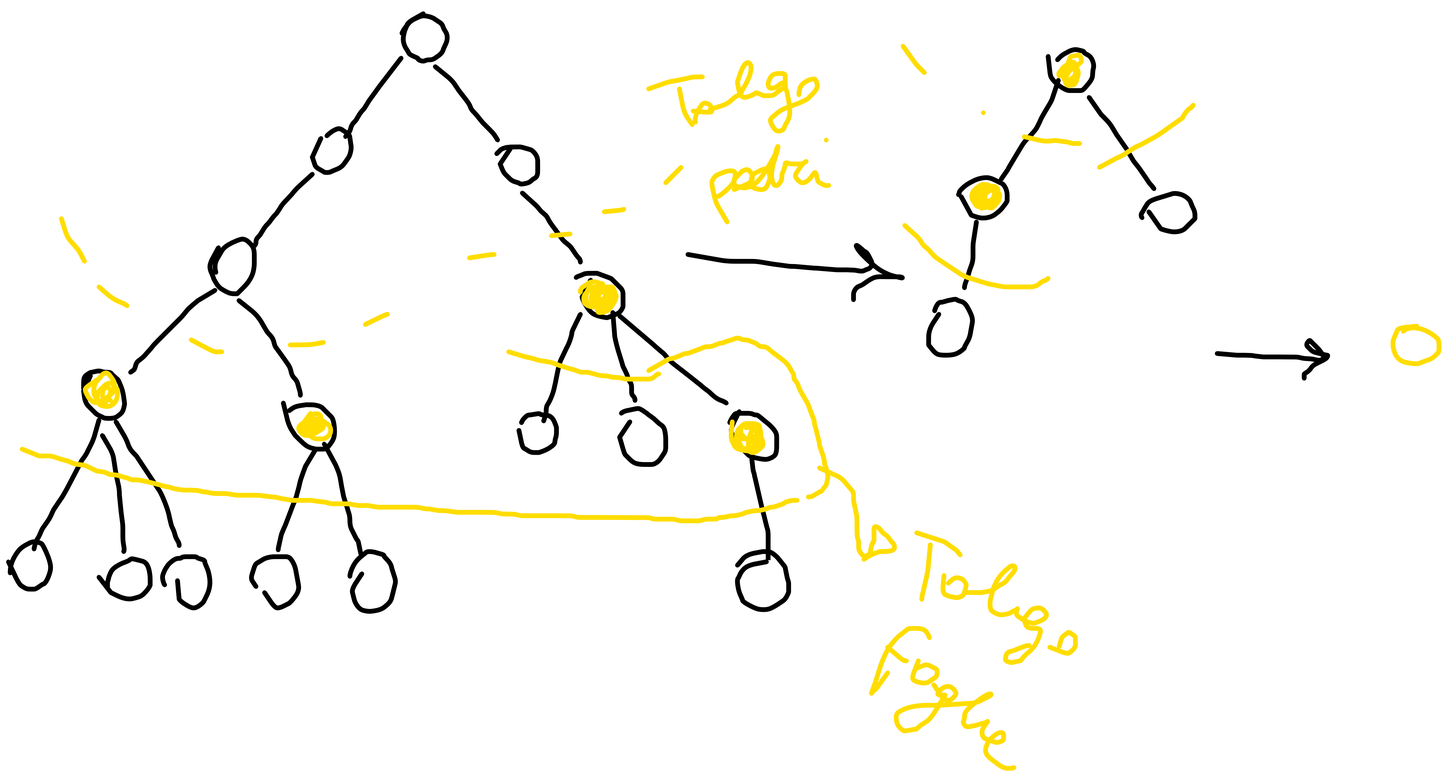
VARI TIPI DI SCAMBI

MIGLIORE



Vogliamo trovare un vertex cover senza foglie:

→ Una procedura che induttivamente dimostri che è sempre che il vertex cover ottimo non contiene nodi foglie



Cover = ∅

WHILE ∃ foglie in G

Aggiungi i padri in Cover

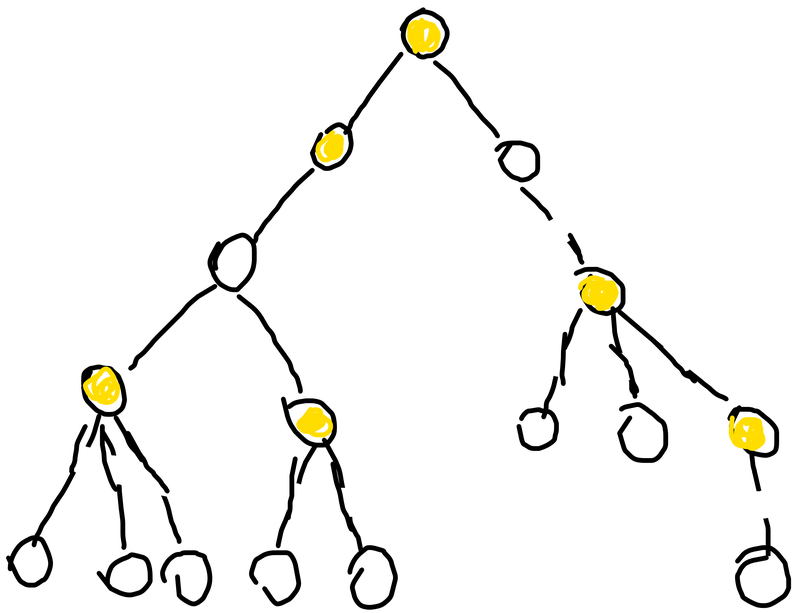
Rimuovi le foglie ed i loro padri da G

RETURN Cover

Questo algoritmo si può sempre applicare?

Sì perché ogni albero finito con $n>1$ ha almeno due foglie

↳ #NODI



Cover = $\emptyset$

WHILE $\exists$ foglie in G

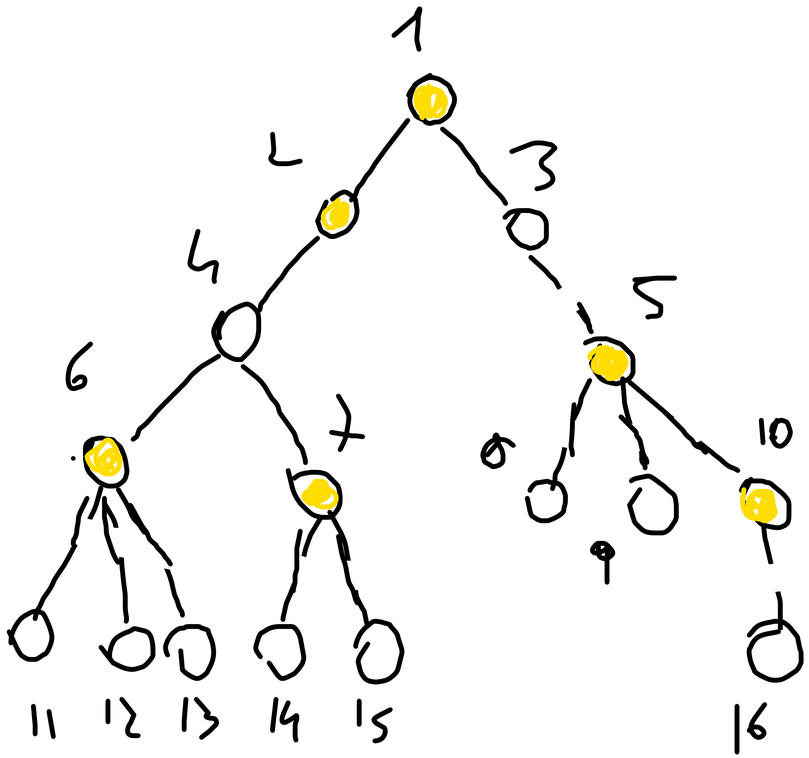
Aggiungi padri in Cover

Rimuovi le foglie ed i loro padri da G

RETURN Cover

Dobbiamo trovare un modo di applicare questo algoritmo in $O(n)$

→ Sappiamo: nei grafi ad albero $|E| = |V| - 1$ ⇒ quello che farei in $O(V+E)$ posso farlo in $O(V)$



Scorro la lista di adiacenze e mi creo due array:

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
P	-	1	1	2	3	4	4	5	5	5	6	6	6	7	7	10
Deg	2	2	2	3	4	4	3	1	1	2	1	1	1	1	1	1

FOGLIE [8 9 11 12 13 14 15 16]

Poi applico questa procedura: FOR-EACH u in F

$S = \emptyset$

padre$(v) = v$

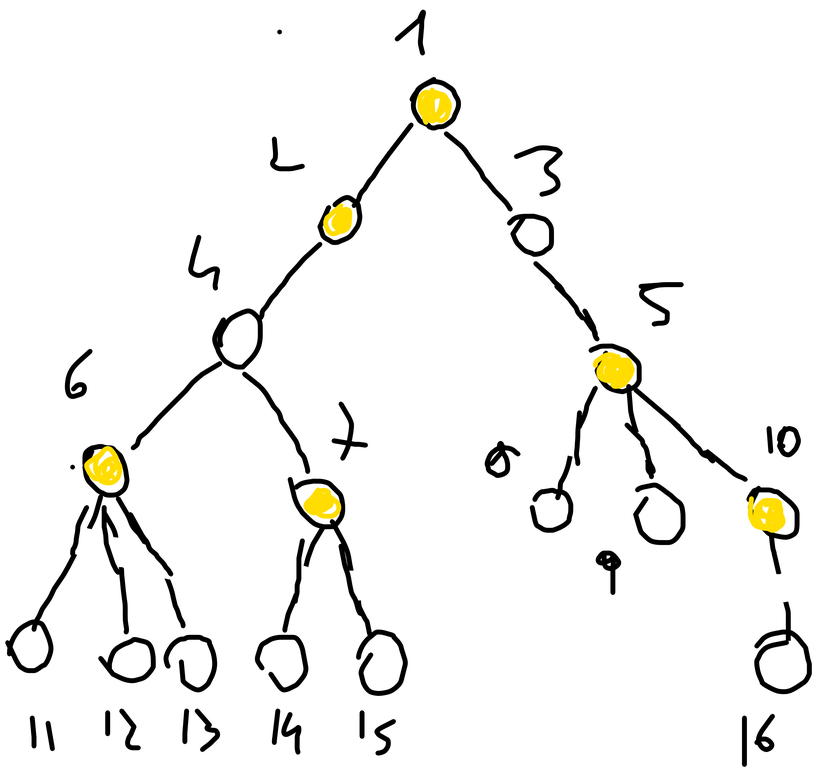
IF deg$(v) = 0$ SKIP (continue)

ELSE deg$(v) = 0$, $S = S \cup \{v\}$

padre$(v) = z$, deg(z) --

IF deg$(z) == 1$

$F = F \cup \{z\}$



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
P	–	1	1	2	3	4	4	5	5	5	6	6	6	7	7	10
Deg	2	2	2	3	4	4	3	1	1	2	1	1	1	1	1	1

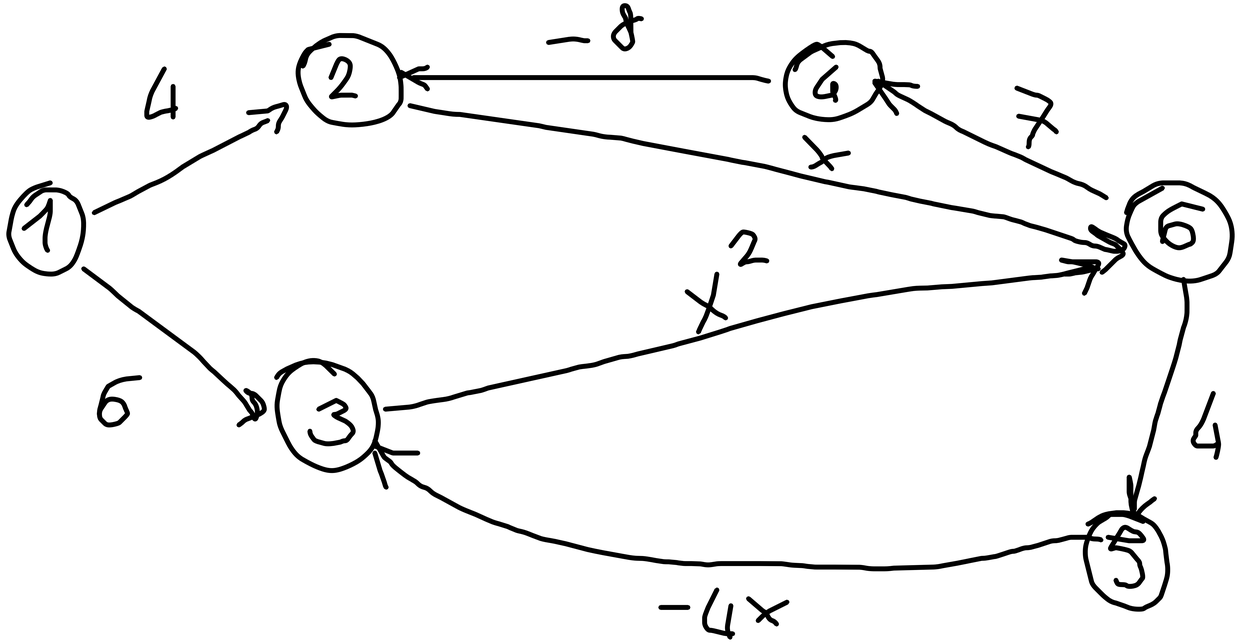
2 0

FOGLIE $[8\ 9\ 11\ 12\ 13\ 14\ 15\ 16]$ < 3

ESERCIZIO

Trovare i valori di x per cui $\exists$ un cammino minimo di lunghezza finita tra il nodo 1 ed il nodo 6.

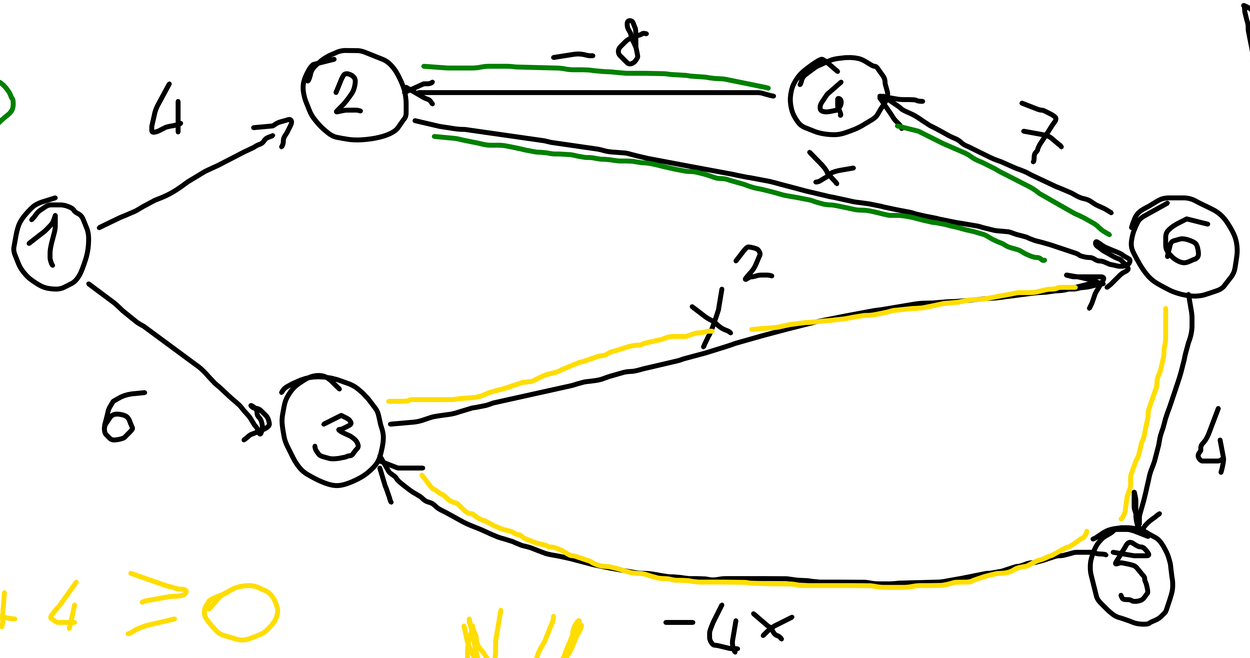
Dato un valore di x adeguato usare l'algoritmo opportuno per calcolare la lunghezza del cammino



ESERCIZIO

Trovare i valori di x per cui ∃ un cammino minimo di lunghezza finita tra il nodo 1 ed il nodo 6.

Dato un valore di x adeguato usare l'algoritmo opportuno per calcolare la lunghezza del cammino



$x - 1 \geq 0$

$x \geq 1$

$x^2 - 4x + 4 \geq 0$

Vero ∀ valore x

Voglio trovare dei valori di x per cui non esistono cicli negativi

$x \geq 1$

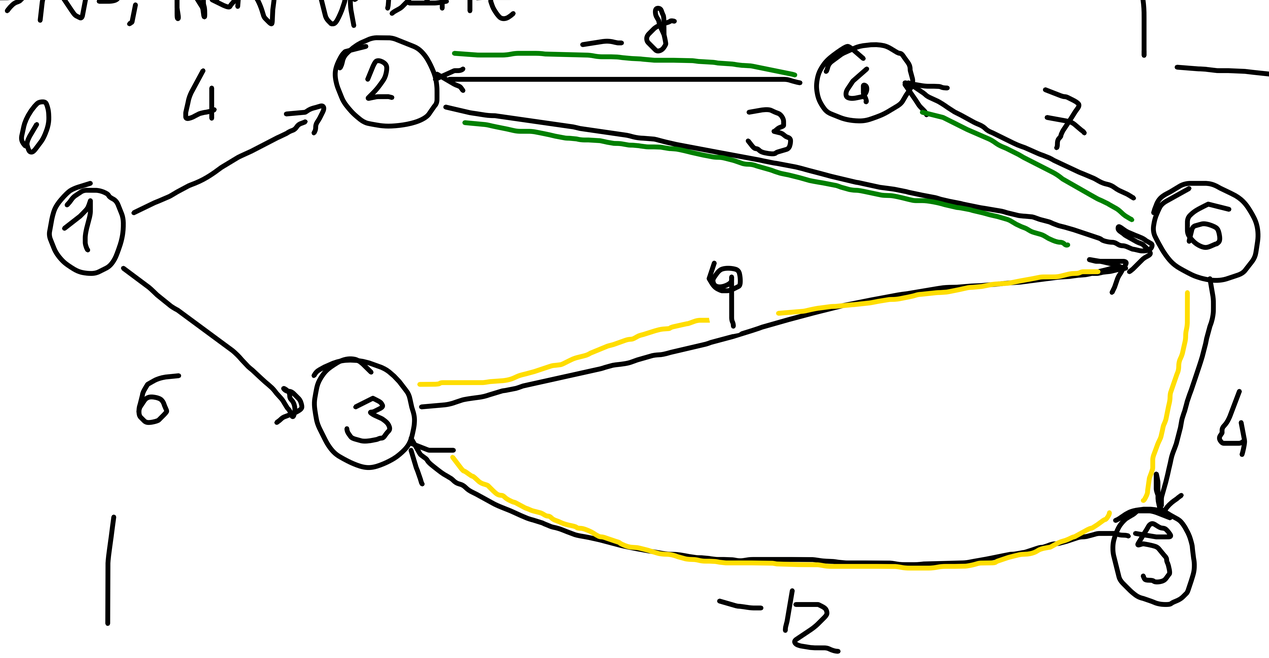
Scelgo $x = 3$ → Per trovare la lunghezza del cammino minimo da 1 a 6 devo utilizzare per forza BELLMAN FORD poiché Dijkstra non è in grado di gestire i pesi negativi.

?

$V.d > u.d + w(u,v)$

↳ SI, UPDATE

↳ NO, NON UPDATE



	1	2	3	4	5	6
It_1	0	~~4~~	~~6~~	~~14~~	~~11~~	~~7~~
It_2	0	4	6	14	11	7