PreIMO 2017 - COMBINATORIA Pomeriggio

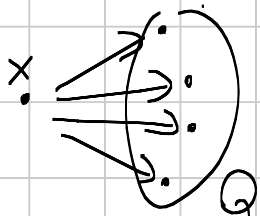
Note Title 25/05/2017

C5

$A, B \quad A \cap B = \emptyset$ t.c. "$A \rightrightarrows B$"

→ quante quadruple? $\binom{60}{4} = \frac{60 \cdot 59 \cdot 58 \cdot 57}{24}$

→ dico che $x \in V(G)$ copre una quadrupla Q se



TESI: posso trovare 4 elementi di $V(G)$ che coprano la quadrupla B; scelgo questi come A.

Come lo dimostro? Pigeonhole.

→ $x \in V(G)$ copre $\binom{d_{out}(x)}{4}$ quadruple.

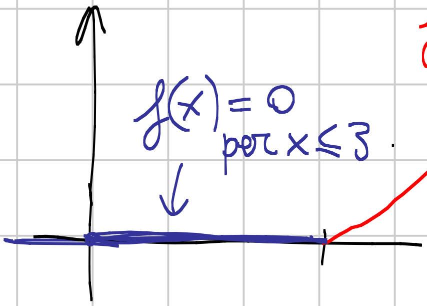
Ma $$\frac{\sum_{x \in V(G)} \binom{d_{out}(x)}{4}}{\binom{60}{4}} \overset{?}{>} 3$$, quindi la tesi. ★

Sì!

So che $\sum_{x \in V(G)} d_{out}(x) = |E(G)| = \binom{60}{2}$

Applico Jensen su questa funzione:

$f(x) = \frac{x(x-1)(x-2)(x-3)}{24}$ per $x > 3$



$$\frac{\star}{60} = \frac{\sum_{x \in V(G)} f(d_{out}(x))}{60\binom{60}{4}} \geq \frac{f\left(\frac{\overbrace{\sum d_{out}(x)}^{\binom{60}{2}}}{60}\right)}{\binom{60}{4}} =$$

$$> \frac{3}{60} \to \text{ok!} \qquad \square$$

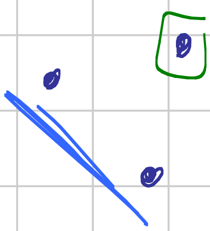
C6

Alberto difende un ordinamento *

Barbara cerca di scoprirlo

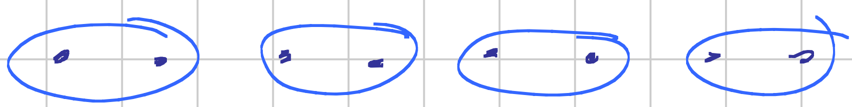
* 20 oggetti

Sol: Si inizia con la difesa



scegliere delle coppie fortunate ~~da difendere~~ ad ogni costo

Per esempio scelgo le coppie in questo modo



quindi $\forall$ richiesta di Barbara
saranno coinvolte almeno 2 coppie
quindi posso sempre evitare di rompere
le coppie

Abbiamo $|I'| \leq 10$

Per l'altro bound devo cercare delle informazioni

B chiede tutte le terne

Caso 1: $\forall$ coppia, la so ordinare

e quindi so tutto!

Caso 2: $\exists$ coppia (a,b) "fortunata" (di cui non so dire l'ordine)

$\Rightarrow \forall x \neq a, b$ vale che

$$\begin{matrix} x > a \\ x > b \end{matrix} \quad \vee \quad \begin{matrix} x < a \\ x < b \end{matrix}$$

$$\underbrace{\text{più piccoli}}_{*} \leq a, b \leq \underbrace{\text{più grandi}}_{*}$$

* queste situazioni sono uguali a quella generale, ma con meno oggetti

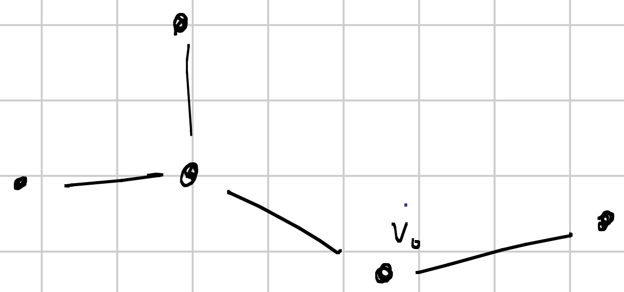
Formalmente "Per induzione su n, se ho n oggetti, ne scopro $\lceil \frac{n}{2} \rceil$ in ordine".

Oppure faccio veramente induzione su $2n$ oggetti, scopro l'ordinamento di $|n-1|$ tra gli altri $2n-2$ (ip. ind.) e aggiungo "a" tra

Quindi $|I'| \geq 10$

C7

C'è un albero

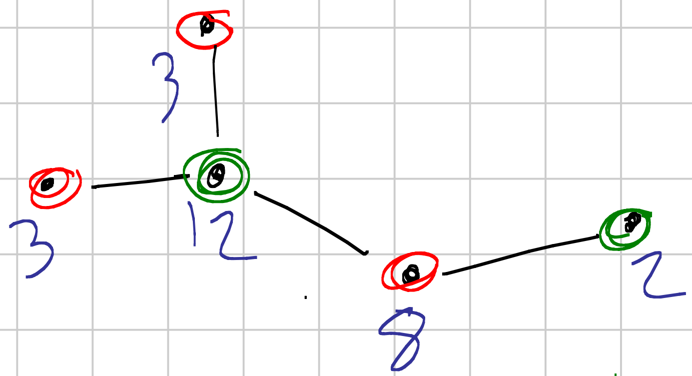


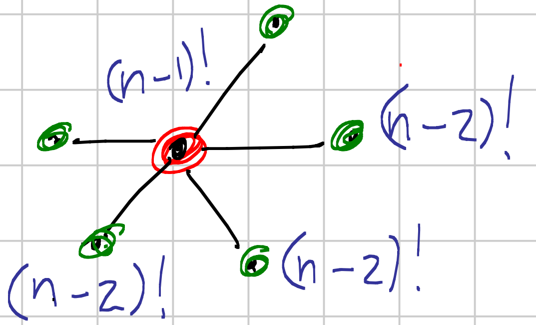
$\forall$ vertice v conto le enumerazioni decrescenti che partono da lui. Sia $f(v)$

Ip: $2016 \mid f(v) \quad \forall v \in V \setminus \{v_0\}$

Tesi: $2016 \mid f(v_0)$.

Sol:





$$(n-1)! = (n-2)! + (n-2)! + \dots + (n-2)!$$

$$f(\text{centrale}) = \sum_{\text{foglia}} f(\text{foglia})$$

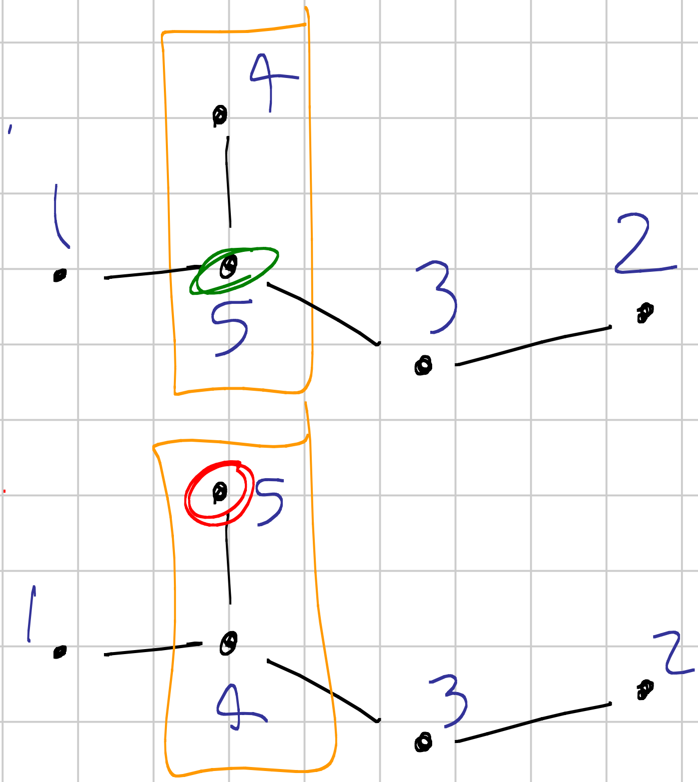
Claim: bipartiziono l'albero e voglio

$$\sum_{\text{rossi}} f(v) = \sum_{\text{verdi}} f(v)$$

$$\| \qquad\qquad\qquad\qquad \|$$

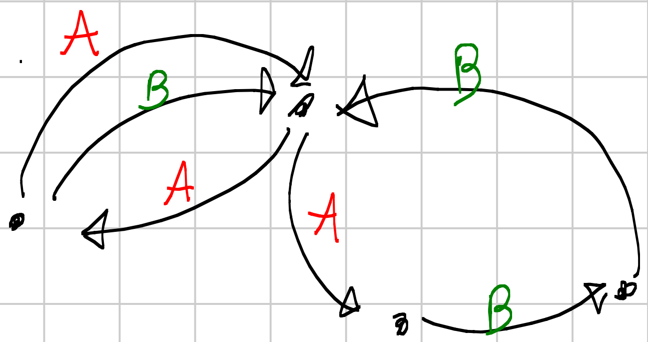
$$\left| \left\{ \begin{array}{l} \text{enumerazioni decr.} \\ \text{che hanno radice} \\ \text{in un rosso} \end{array} \right\} \right| \qquad = \qquad \text{verde}$$

esibisco una bigezione tra i 2 insiemi



gli altri stanno fermi e non fanno danni

C 8 C'è un grafo orientato ed etichettato

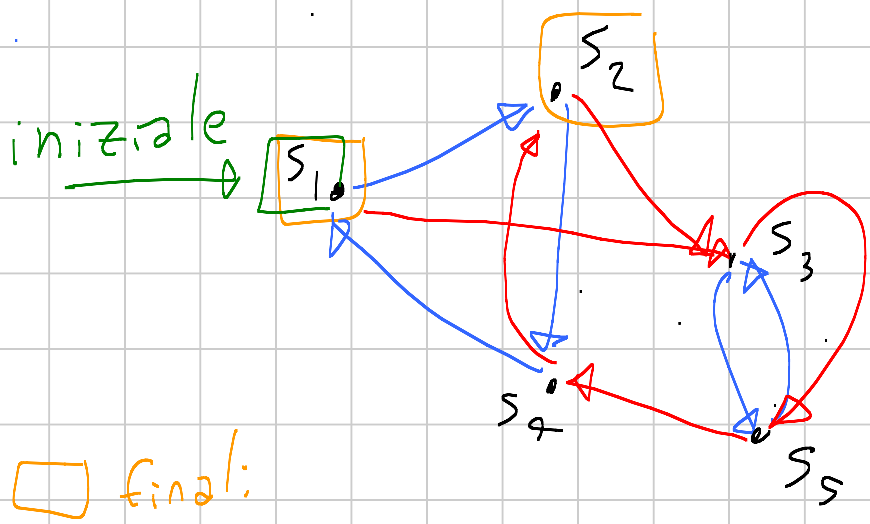


Ip: tutte le parole lunghe 2^n sono implementabili

Th: tutte le parole finite sono implementabili

Automi a stati finiti e Pumping Lemma

Def: ASF è un grafo di "stati" in cui



- c'è uno stato iniziale
- ci sono degli stati finali

C'è anche Σ un alfabeto

$\Sigma = \{A, B\}$

· e una "funzione di transizione"

$$F : S \times \Sigma \longrightarrow S$$
$$(s, \ell) \longmapsto \tilde{s}$$

Data una parola con lettere $\subseteq \Sigma$ posso camminare lungo il grafo di stati

Thm (Pumping Lemma):

Supponiamo che ci sia ω una parola di lunghezza $> |S|$ accettata dall'ASF,

allora $\omega = x\,y\,z$ dove x, y, z sono parole

y non è la parola nulla
e xy ha lunghezza $\leq |S|+1$

e anche la famiglia di parole $x y^n z$ è accettata dall'ASF $\forall\, n \in \mathbb{N}$

Dim: Per pigeonhole, entro la $|S|+1$-esima lettera vedo 2 volte uno stesso stato s'
quindi: divido ω in questo modo

$$\omega = x \uparrow y \uparrow z$$

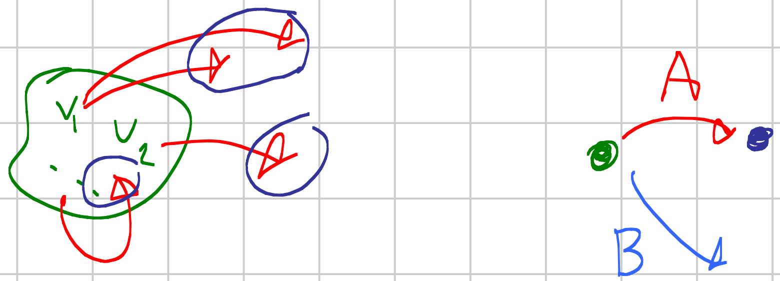
s' s'

ora tutte le stringhe $xy\dots yz$ sono accettate.

Sol del C8.a

Oss: il 2^n mi ricorda i sottoinsiemi dei vertici

Costruisco un nuovo grafo: i vertici sono i sottoinsiemi di quello di partenza e pongo questi archi.



Oss: c'è lo stato iniziale (tutto) e $\forall$ stato c'è esattamente una freccia con la A e una con la B

L'ipotesi diventa questa: per ogni parola di lunghezza 2^n, finisco in un vertice $\neq \emptyset$

1° modo: mostro la tesi per induzione sulla lunghezza della parola e seguo un ragionamento come prima.

2° modo: Definisco l'ASF con il grafo di prima dove {tutto} è lo stato iniziale, $\emptyset$ è l'unico stato non finale

ora una parola lunga $k+1$ si spezza in 3 parti e posso levare quella intermedia per ottenerne una valida.

3° modo: Per assurdo ...

C 8.b Qui il grafo è non diretto ma la lunghezza che conosco è solo $2n$.

Sol:

Oss. non è vero che

$\_\_ \omega \_\_ AA$ è una delle più corte implementabili, altrimenti

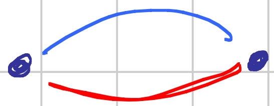
ωA lo è, torno indietro

al penultimo vertice e ottengo ωAA

Oss: neanche $\omega ABBA$ è una delle più corte non implementabili:

Spero che saper fare $ABAB\ldots AB$ mi basti per fare tutto.

Oss: esistono dei grafi facili con i quali implemento tutto:



Se ho un sottografo per cui da ogni suo vertice, rimango dentro con passi del tipo A o B, allora implemento tutte le parole.

Ora prendiamo la parola $(AB)^n = \underbrace{AB \ldots AB}_{n \text{ volte}}$

adesso osservo che vedo 3 volte (almeno) lo stesso vertice (pigeonhole)

$$AB|AB\,A|B \;\ldots\ldots\; AB|AB$$

3 volte lo stesso

c'è una sottostringa pari

ora voglio mostrare che i vertici visitati qui formano un grafo del tipo detto

infatti ognuno dei vertici vede una A e una B compreso quello iniziale (e finale).