Classi problemi → P (Polinomiale)

- Es. somma/differenza (che sai misurare e risolvere)

NP (Non Polinomiali)

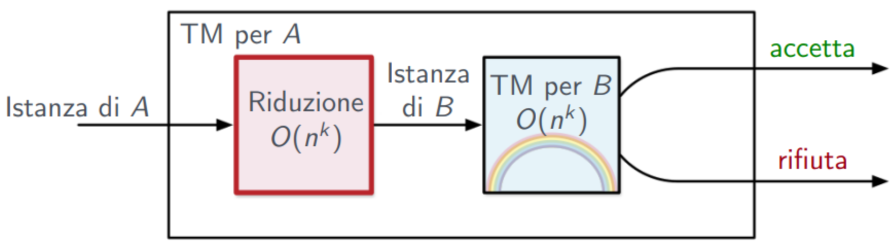
- Classi problemi noti essere difficili da risolvere efficientemente

| | P | NP |
|---|---|---|
| Facili da risolvere | ✓ | ? |
| Facili da verificare | ✓ | ✓ |

Tipi di problemi in NP:

- Set partitioning
  - $S_1 + S_2 = S$
  - Due sottoinsiemi disgiunti (aka diversi) tali che la somma degli elementi di $S_1$ è uguale alla somma di $S_2$
    - $5 + 5 = 10$
    - Un 5 è dato da $1 + 1 + 1 + 1 + 1$
    - Un altro 5 è dato da $2 + 3$
- Clausole booleane
  - SAT = soddisfacibilità
    - E.g. $(a \land b \land c) \ldots = 1$ (vero = va bene)
  - Varianti = 3-SAT – k-SAT
- Grafi
  - Hamiltoniano = attraversa tutti gli archi una sola volta
  - Vertex Cover = copertura dei vertici partendo da una sorgente
  - Independent Set = insieme indipendente di vertici = vertici non vicini ma coprono tutto il grafo
  - Color = "Colorare" tutto il grafo attraversandone i vertici

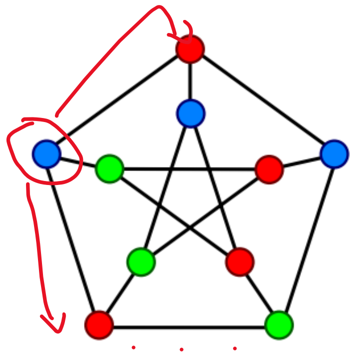
Se $A \leq_P B$, e $B \in P$, allora $A \in P$:



**3. (12 punti)** Una 3-colorazione di un grafo non orientato $G$ è una funzione che assegna a ciascu vertice di $G$ un "colore" preso dall'insieme $\{R, G, B\}$, in modo tale che per qualsiasi arco $\{u, v\}$ colori associati ai vertici $u$ e $v$ sono diversi. Una 3-colorazione è *equa* se per ogni coppia di colori, i numero di nodi associati a ciascun colore differisce di al più 1.

EQUITABLE-3-COLOR è il problema di trovare una 3-colorazione equa:

$$\text{EQUITABLE-3-COLOR} = \{\langle G \rangle \mid G \text{ è un grafo che ammette una 3-colorazione equa}\}$$



Esempio di colorazione equa con 4 vertici rossi, 3 vertici blu e 3 vertici verdi.

(a) Dimostra che EQUITABLE-3-COLOR è un problema NP
(b) Dimostra che EQUITABLE-3-COLOR è NP-hard, usando 3-COLOR come problema NP-hard d riferimento.

a)

Verifica = Vedere se valgono tutte le condizioni

Esiste un verificatore che ha un "certificato" (input che mi permette di verificare questo)

- Certificato = $Gr$, grafo con $\{R, G, B\}$ colori
- Verificatore
  - Controlla che esistano tutti i vertici da 1 ad $n$
  - Parto da un vertice e verifico che per l'arco a cui è connesso
    - $v$ allora tramite arco $\{u, v\}$ arrivo ad una coppia di colori diversi
    - i colori differiscono al più 1
  - Lo faccio per tutti i vertici

Problema in NP = verificabile ma non risolvibile

NP-hard = se esiste una soluzione efficiente, risolvo tutti i problemi del tipo

➔ $P = NP$

b)

$$A \leq_m B$$

Applicata al nostro problema, dove $A$ è un superinput, e $B$ è un sottoinput

$$3 - COLOR \leq_m EQUITABLE - 3 - COLOR$$

(⇒) Usiamo EQUITABLE per dimostrare che siamo in 3-COLOR

- Verifico che sia una 3-colorazione equa
- Ho il mio grafo, percorro per ogni arco una coppia di colori
- Percorro tutto il grafo

- Se, alla fine del grafo, ottengo almeno un colore diverso, questo fa parte di almeno un'altra 3-colorazione (siamo ancora dentro al problema A, ma abbiamo usato B)

(⇐) Usiamo 3-COLOR per dimostrare che abbiamo usato EQUITABLE

- Abbiamo una 3-colorazione equa
- Questa viene ottenuta da una 3-colorazione
    - Percorro tutto il grafo
    - Per ogni vertice, uso una coppia di archi e verifico che tutti i vertici siano colorati diversamente
    - Se questo vale, allora funziona
- Funziona di sicuro, dato che "equo" dipende da 3-colorazione

(Consiglio: Leggere le soluzioni di Filè (pre 2019-2020 -per avere delle soluzioni testuali fattibili)

**2.** *Dimostra che il seguente linguaggio è indecidibile:*

$$A_{1010} = \{\langle M \rangle \mid M \text{ è una TM tale che } 1010 \in L(M)\}.$$

$$A \leq_m B$$

Come siamo messi ora:

$$A \leq_m A_{1010}$$

Usare un problema indecidibile al posto di $A$:

$$A_{TM} - E_{TM} - EQ_{TM} - HALT_{TM}$$

Soluzione base:

$$A_{TM} \leq_m A_{1010}$$

La funzione di riduzione $f$ è calcolata dalla seguente macchina di Turi

$F =$ "su input $\langle M, w \rangle$, dove $M$ è una TM e $w$ una stringa:

1. Costruisci la seguente macchina $M_w$:
   $M_w =$ "su input $x$:
    1. Se $x \neq 1010$, rifiuta.
    2. Se $x = 1010$, esegue $M$ su input $w$.
    3. Se $M$ accetta, *accetta*.
    4. Se $M$ rifiuta, *rifiuta*."
2. Restituisci $\langle M_w \rangle$."

$M$ = macchina che risolve $A_{TM}$

Nel momento in cui abbiamo "risolto" $A_{1010}$ comincio ad eseguire $M$ per risolvere $A_{TM}$

Se accetta, accetta (e viceversa)

**3. (9 punti)** Una *Turing machine con alfabeto ternario* è una macchina di Turing deterministica a singolo nastro dove l'alfabeto di input è $\Sigma = \{0, 1, 2\}$ e l'alfabeto del nastro è $\Gamma = \{0, 1, 2, \sqcup\}$. Questo significa che la macchina può scrivere sul nastro solo i simboli $0, 1, 2$ e blank: non può usare altri simboli né marcare i simboli sul nastro.

Dimostra che ogni linguaggio Turing-riconoscibile sull'alfabeto $\{0, 1, 2\}$ può essere riconosciuto da una Turing machine con alfabeto ternario.

$$TM_{Nastro\ Singolo} \leq_m TM_{Alfabeto\ Ternario}$$

$$Funzione\ di\ transizione = \delta\ (delta): Q\ x\ \Gamma\ (\text{Gamma}) \mapsto \text{Q x } \Gamma \text{ x } \{\text{L, R}\}$$

(⇒) Con la TM a nastro singolo, possiamo semplicemente provare ad inserire dei simboli che, codificando esattamente $\{0,1,2\}$ riescono a garantire che la sua normale funzione di transizione "stampi" esattamente quei simboli. Ci muoviamo a sinistra e a destra svolgendo tutte le transizioni codificando solo quei simboli. Se becchiamo simboli "fuori dal vaso", allora rifiuta, altrimenti accetta

(⇐) Con la TM ad alfabeto ternario abbiamo codificato, seguendo l'alfabeto, esattamente tre simboli. Rappresenta una variante "banale" della TM a nastro singolo, dato che fa esattamente lo stesso, "ma non solo con 3 simboli, ma qualsivoglia simbolo prenda = tutto, allora va sempre bene".