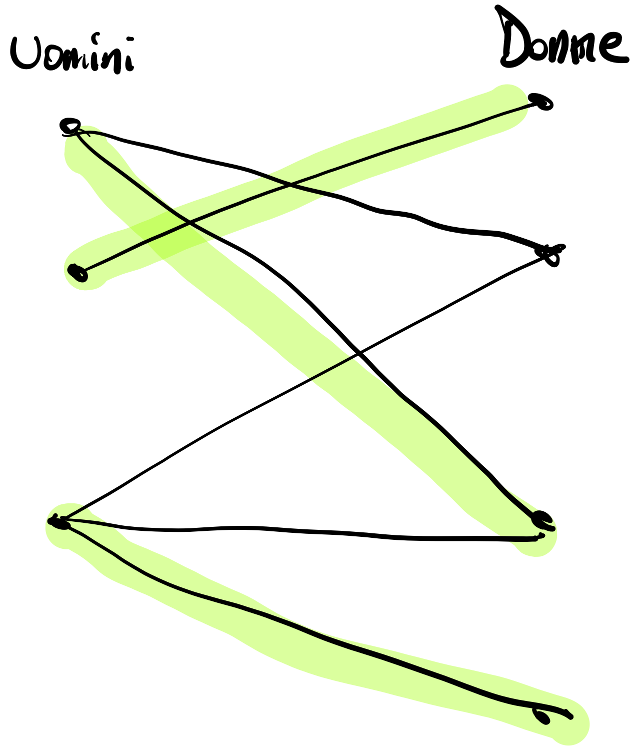
PROBLEMA DI MAX MATCHING

INPUT : $G=(V,E)$ non orientato
<u>bipartito</u>

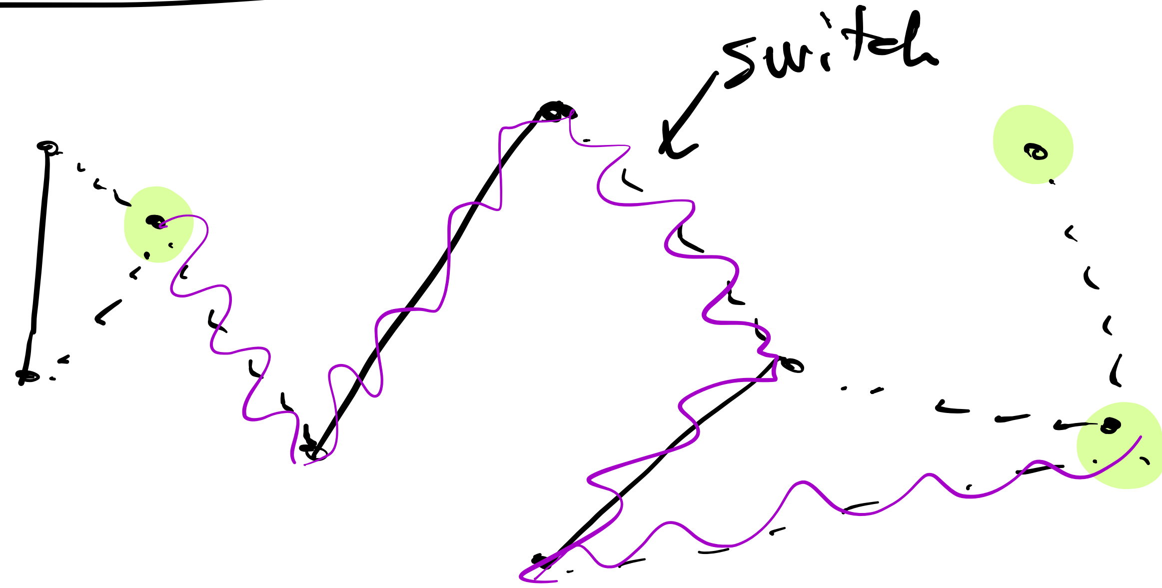
SOL. AMMISSIBILI: Matching M
(scelta di lati per cui nessun vertice incida su >1 lato)

FUN. OBIETTIVO: $|M|$

TIPO : MAX



ALGORITMO DEL CAMMINO AUMENTANTE



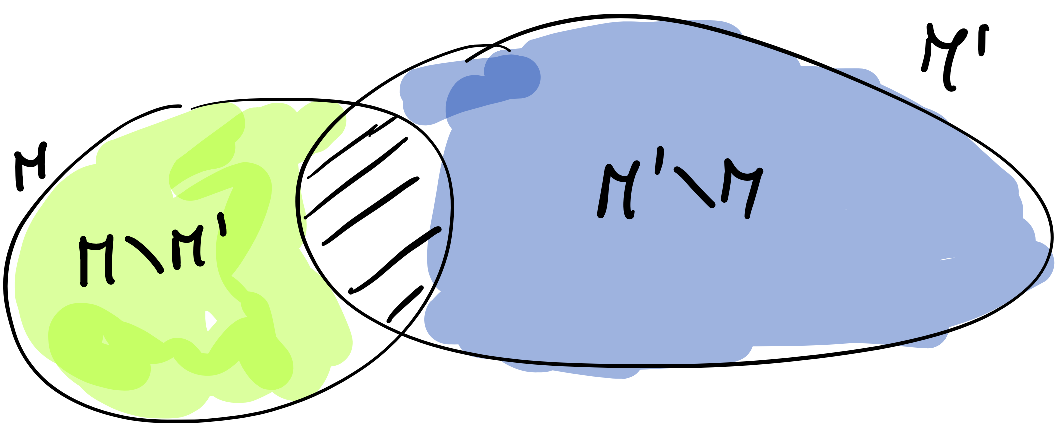
G con matching M

vertice esposto = vertice su cui non incide nessun lato del matching augmenting path = cammino che parte e arriva su un vertice esposto e alterna lati liberi e lati presi

Lemma 1: Se esiste un cammino aumentante per il matching M, M non è massimo

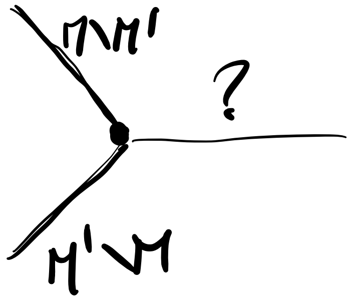
Lemma 2: Se M non è massimo, esiste un cammino aumentante

Dim: Sia M' un altro matching con $|M'| > |M|$.

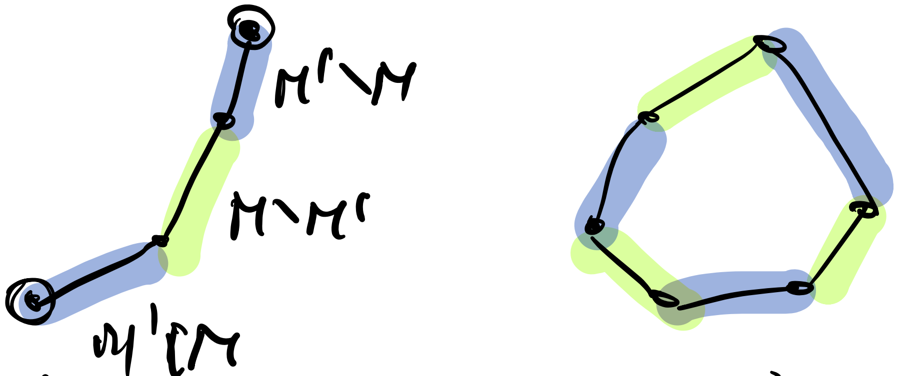


$M \Delta M' = (M \setminus M') \cup (M' \setminus M)$

• Nessun vertice può avere più di due lati incidenti in $M \Delta M'$



$M \Delta M'$



$\exists$ almeno un cammino che inizia e finisce con $M' \setminus M$

$\Rightarrow$ cammino aumentante per M

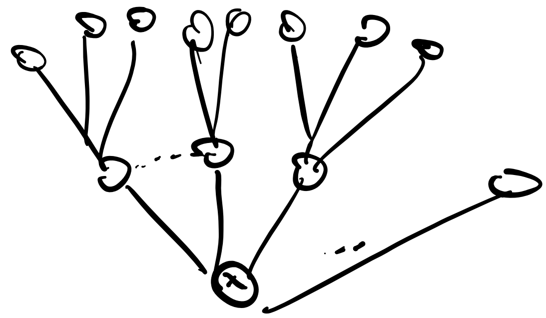
$\square$

VISITE DI GRAFI

nodi
- sconosciuti (BIANCHI)
- conosciuti ma non visitati [FRONTIERA] (GRIGI)
- visitati (NERI)

```
∀x  c(x) ← W
F ← {x_seed}
c(x_seed) ← G
while F ≠ ∅ :
    x ← pick(F)
    visit(x)
    c(x) ← B
    for y in neighbor(x) :
        if c(y) == W :
            F ← F ∪ {y}
            c(y) = G
```

pick(F):
- STACK: DFS
- QUEUE: BFS
- ...



$O\left(\frac{n^2}{2} n\right)$

$\frac{n}{2}$

FIND AUGMENTING (G, M)

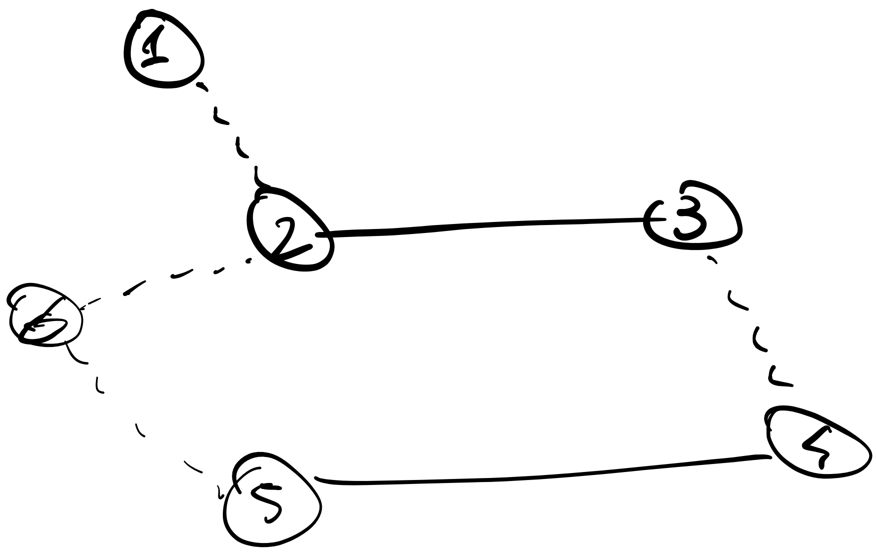
$O(nm)$

$X \leftarrow$ vertici esposti in M

$\forall x \in X$

BFS(x) ⟵ con alternanza di lati di M e lati fuori da M

PERCHÉ LA BFS NON FUNZIONA SU GRAFI NON BIPARTITI



Teorema: Bipartite Max Matching $\in P$

Corollario: Perfect Matching $\in P$

PROBLEMA DI LOAD BALANCING

INPUT:

$m > 0$ numero di macchine

$n > 0$ numero di task

$(t_i)_{i \in n}$ durate dei task

$t_i > 0$

OUTPUT: $\alpha : n \to m$

L

$$L_j \triangleq \sum_{i : \alpha(i) = j} t_i$$

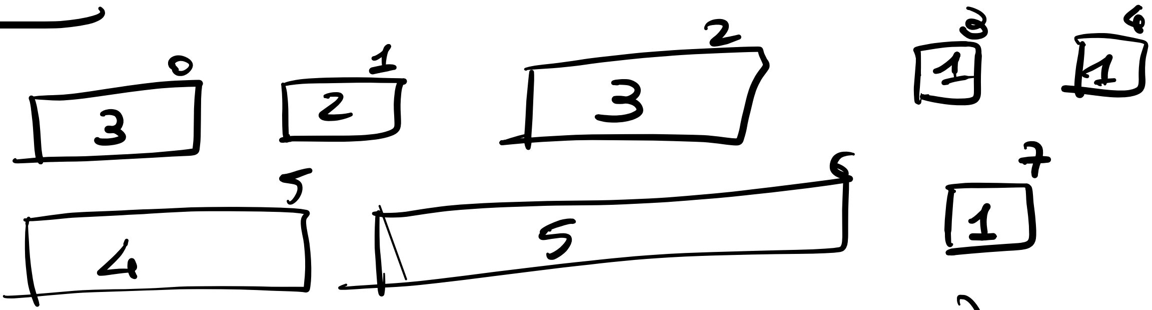
$$L \triangleq \max_j L_j$$

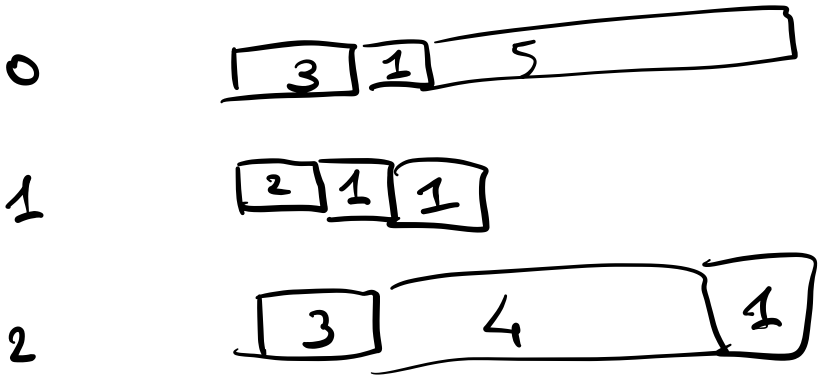
FUNZ. OBIETTIVO: L

TIPO: MIN

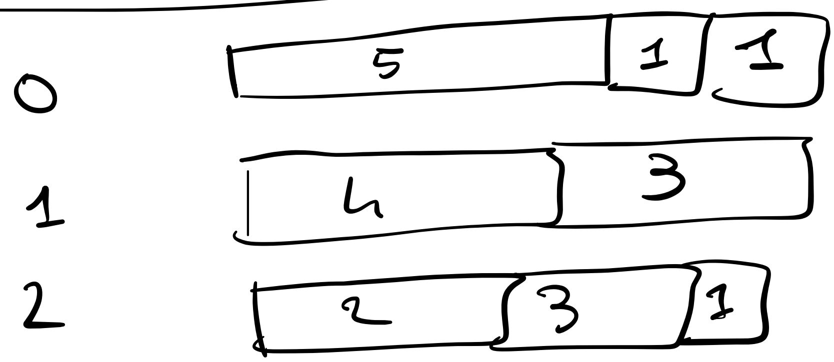
ESEMPIO $n = 8$ $m = 3$

$\sum_j t_j = 20$





$$\left.\begin{array}{l} L_0 = 9 \\ L_1 = 3 \\ L_2 = 8 \end{array}\right\} L = 9$$



$$\left.\begin{array}{l} L_0 = 7 \\ L_1 = 7 \\ L_2 = 6 \end{array}\right\} L = 7$$

Teorema: LOAD BALANCING $\in NPO_c$

Def: Un p.o. Π è $\in NPO_c$ sse $\Pi \in NPO$ e $\hat{\Pi} \in NP_c$

ALGORITMO GREEDY PER LOAD BALANCING

$O(nm)$

$A_i \leftarrow \emptyset \quad \forall i \in m$

$L_i \leftarrow 0 \quad \forall i \in m$

for $j = 0, \dots, m-1$

$\quad \bar{i} \leftarrow \arg\min_i L_i$

$\quad A_{\bar{i}} \leftarrow A_{\bar{i}} \cup \{j\}$

$\quad L_{\bar{i}} \leftarrow L_{\bar{i}} + t_j$

α assegna ogni elemento di A_i a i

Teorema: GREEDY BALANCE è una 2-Approssimazione per LOAD BALANCING

Dim: Chiamiamo L^* la soluzione ottima.

$\boxed{\text{Oss. 1}}$ $\quad L^* \geq \underbrace{\frac{1}{m} \sum_j t_j}$

$L_i < \frac{1}{m} \sum_j t_j \quad \forall i$

$\sum_i L_i < \sum_j t_j \quad$ IMP.

$\boxed{\text{Oss. 2}}$ $L^* \geq \max_j t_j$

Sia $\hat{i}$ t.c. $L_{\hat{i}} = L$ e sia $\hat{j}$ l'ultimo compito che le è stato assegnato.

$$L_{\hat{i}} - t_{\hat{j}} \leq \underbrace{L_i'}_{\text{carico in quel momento}} \leq L_i \quad \forall i$$

$$L_{\hat{i}} - t_{\hat{j}} \leq L_i \quad \forall i \in m$$

$$m\left(L_{\hat{i}} - t_{\hat{j}}\right) \leq \sum_i L_i = \sum_j t_j$$

(A) $$L_{\hat{i}} - t_{\hat{j}} \leq \frac{1}{m} \sum_j t_j \underset{\text{oss 1.}}{\leq} L^*$$

$$L = L_{\hat{i}} = \underbrace{L_{\hat{i}} - t_{\hat{j}}}_{\substack{\leq L^* \\ (A)}} + \underbrace{t_{\hat{j}}}_{\substack{\leq L^* \\ \boxed{\text{Oss.2}}}} \leq 2L^*$$

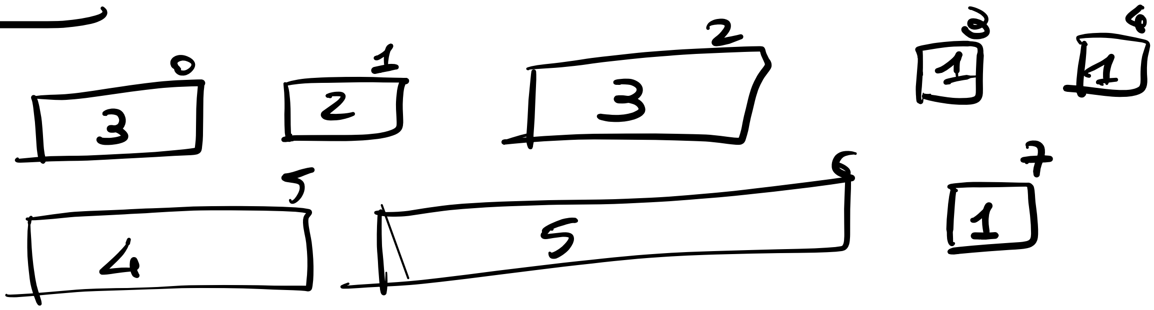
$$\boxed{\frac{L}{L^*} \leq 2}$$

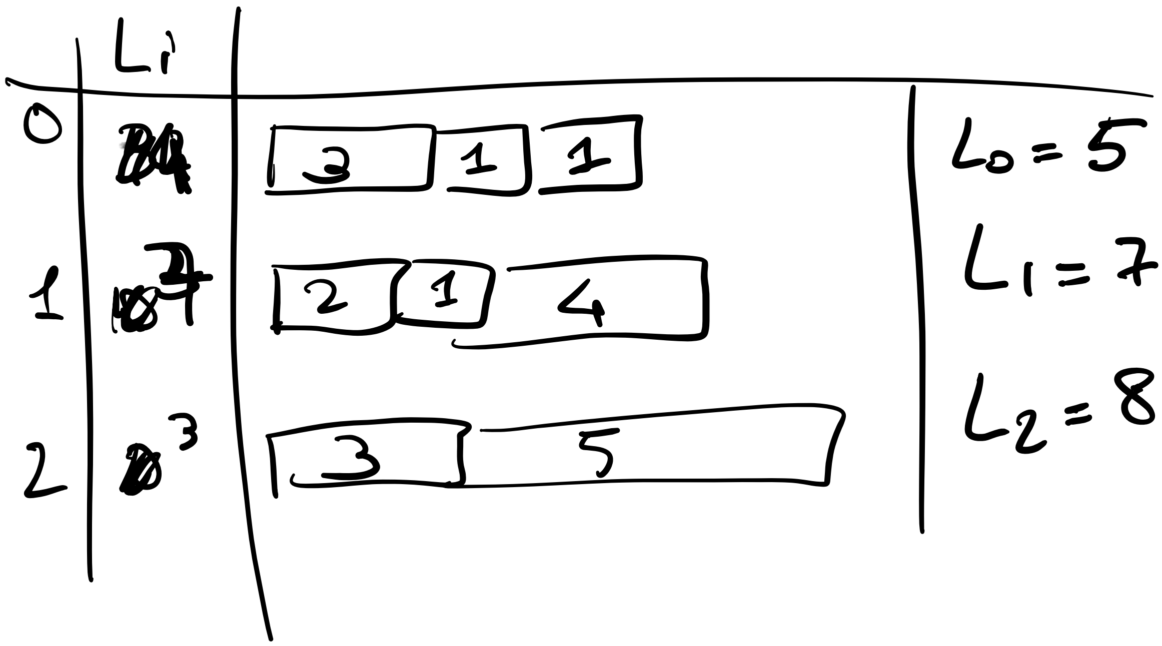
$\square$

ESEMPIO

$n = 8$ $m = 3$

$\sum_j t_j = 20$





$L_0 = 5$

$L_1 = 7$

$L_2 = 8$

$L = 8$