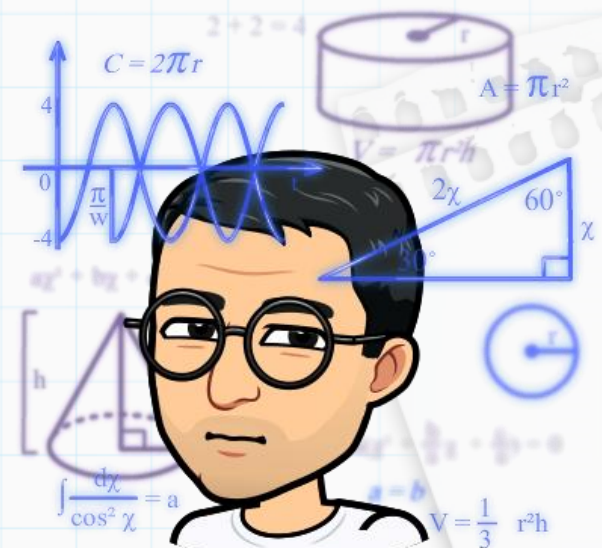
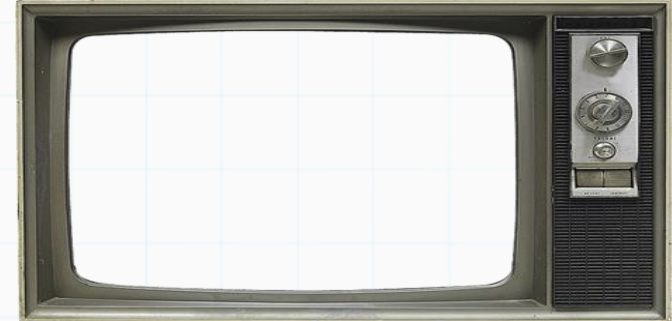


Programação Inteira

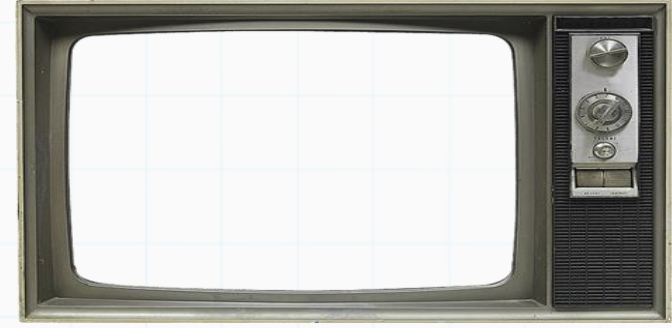
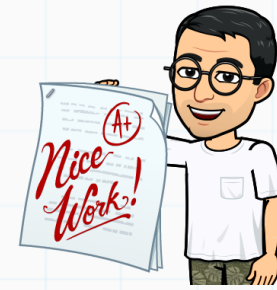
Professor : Yuri Frota

www.ic.uff.br/~yuri/pi.html

yuri@ic.uff.br



Tarefa de Casa



No trabalho anterior, implementamos o modelo de coloração descrito a seguir:

$$\min \sum_{j=1}^n w_j \quad (1)$$

Vamos agora transformar este branch-and-bound num branch-and-cut colocando um “callback” de separação de corte de usuário (user cut), i.e. um corte para fortificar a formulação

$$\sum_{j=1}^n x_{ij} = 1, \quad \forall i \in V \quad (2)$$

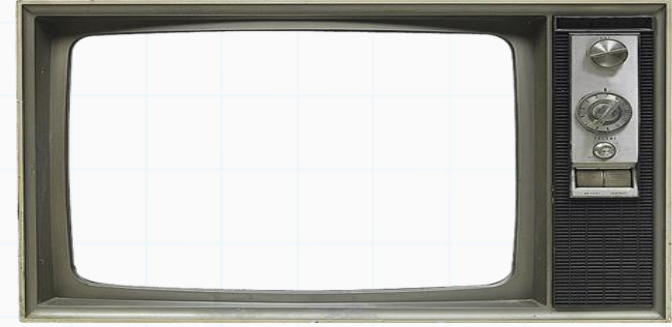
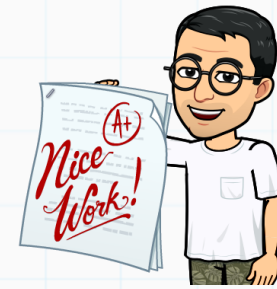
$$x_{ij} + x_{kj} \leq w_j, \quad \forall (i, k) \in E, \quad 1 \leq j \leq n \quad (3)$$

$$w_j \geq w_{j+1}, \quad \forall 1 \leq j \leq n-1 \quad (4)$$

$$w_j \leq \sum_{i \in V} x_{ij}, \quad \forall 1 \leq j \leq n \quad (5)$$

$$x_{ij} \in \{0, 1\}, \quad \forall i \in V, \quad 1 \leq j \leq n \quad (6)$$

Tarefa de Casa

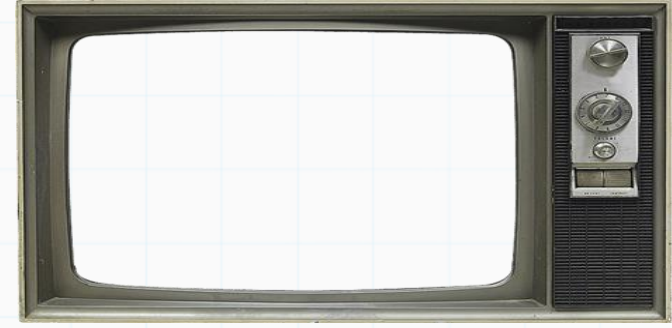
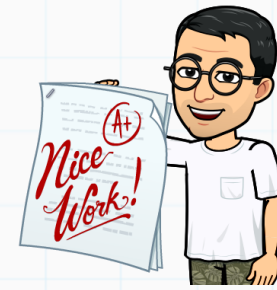


Vamos gerar o corte clique:

$$\sum_{i \in S} x_{ij} \leq w_j, \quad \forall \text{ clique } S \subseteq V, \quad \forall 1 \leq j \leq n$$

Repare que este é um corte não necessário para o modelo estar completo/correto (user cut), logo ele é usado para cortar soluções fracionárias

Tarefa de Casa

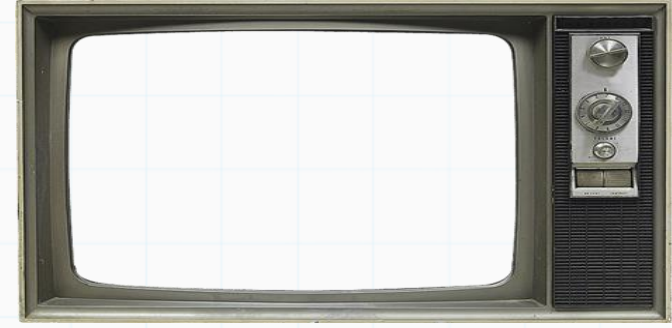
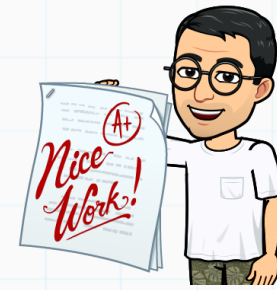


Vamos gerar o corte clique:

$$\sum_{i \in S} x_{ij} \leq w_j, \quad \forall \text{ clique } S \subseteq V, \quad \forall 1 \leq j \leq n$$

- Não vamos gerar todos os cortes, somente aqueles que violados pela solução fracionária corrente.
- O método de separação deverá procurar (heurísticamente) uma clique maximal que esteja violada.

Tarefa de Casa



Vamos gerar o corte clique:

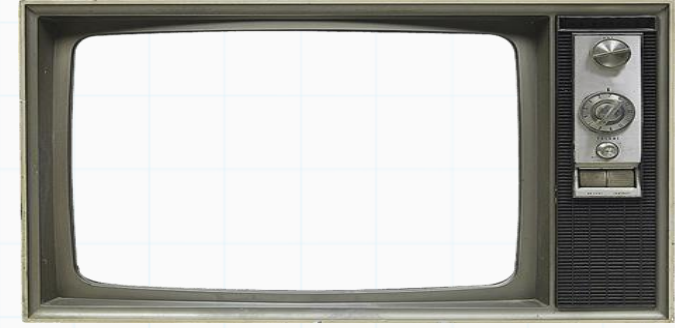
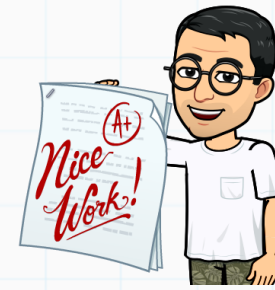
$$\sum_{i \in S} x_{ij} \leq w_j, \quad \forall \text{ clique } S \subseteq V, \quad \forall 1 \leq j \leq n$$

- Não vamos gerar todos os cortes, somente aqueles que violados pela solução fracionária corrente.
- O método de separação deverá procurar (heuristicamente) uma clique maximal que esteja violada.
- Para isso, dada uma solução fracionária (x^*, w^*) o método deve procurar por uma cor j e clique S onde :

$$\sum_{i \in S} x_{ij}^* > w_j^*$$

para inserir os respectivos cortes (user cuts) no modelo.

Tarefa de Casa



Vamos gerar o corte clique:

$$\sum_{i \in S} x_{ij} \leq w_j, \quad \forall \text{ clique } S \subseteq V, \quad \forall 1 \leq j \leq n$$

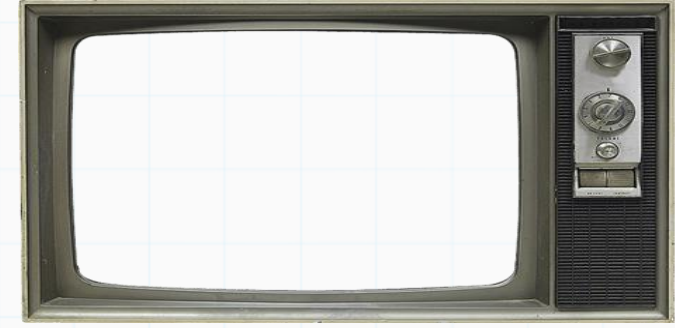
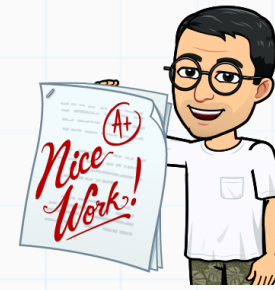
- Não vamos gerar todos os cortes, somente aqueles que violados pela solução fracionária corrente.
- O método de separação deverá procurar (heurísticamente) uma clique maximal que esteja violada.
- Para isso, dada uma solução fracionária (x^*, w^*) o método deve procurar por uma cor j e clique S onde :

$$\sum_{i \in S} x_{ij}^* > w_j^* + \epsilon$$

para inserir os respectivos cortes (user cuts) no modelo.

Erro numérico ou o quanto
tem que estar violada para ser
inserida

Tarefa de Casa



Vamos gerar o corte clique:

$$\sum_{i \in S} x_{ij} \leq w_j, \quad \forall \text{ clique } S \subseteq V, \quad \forall 1 \leq j \leq n$$

- Não vamos gerar todos os cortes, somente aqueles que violados pela solução fracionária corrente.
- O método de separação deverá procurar (heurísticamente) uma clique maximal que esteja violada.
- Para isso, dada uma solução fracionária (x^*, w^*) o método deve procurar por uma cor j e clique S onde :

$$\sum_{i \in S} x_{ij}^* > w_j^*$$

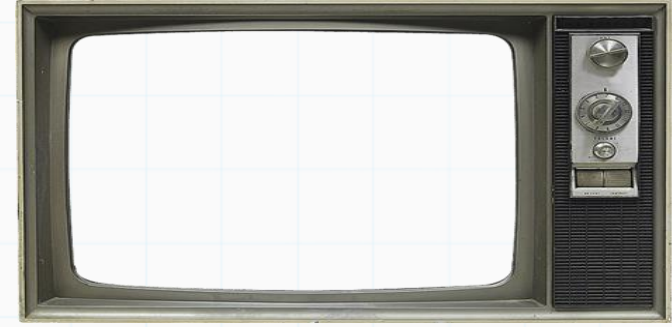
para inserir os respectivos cortes (user cuts) no modelo.

Na verdade, este problema de separação de corte é o problema de achar a clique ponderada máxima

$$\max_j \sum_{u \in V} x_{uj}^* z_u$$

$$\begin{aligned} z_u + z_v &\leq 1, & \forall (u, v) \in E \\ z_u &\in \{0, 1\}, & \forall u \in V \end{aligned}$$

Tarefa de Casa



Ex:

- Na função de “usercut” do “callback”, vamos pegar as variáveis do problema.

$$w_1=1$$

$$w_2=1$$

$$w_3=0.5$$

$$x_{11}=0.3$$

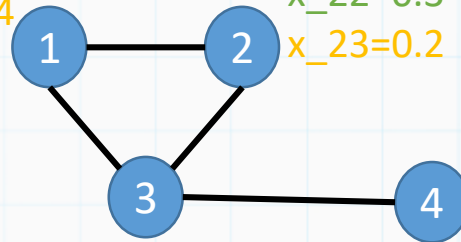
$$x_{12}=0.3$$

$$x_{13}=0.4$$

$$x_{21}=0.5$$

$$x_{22}=0.3$$

$$x_{23}=0.2$$



$$x_{31}=0.3$$

$$x_{32}=0.7$$

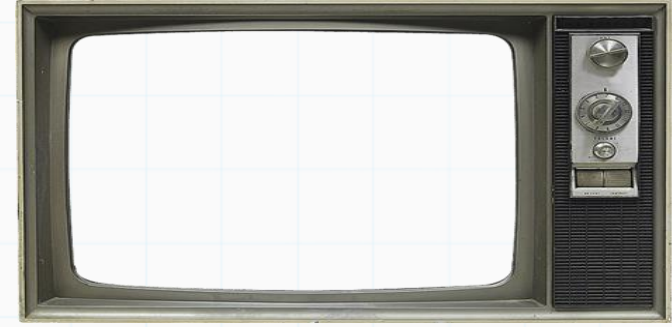
$$x_{41}=0.7$$

$$x_{42}=0.3$$

Veja que as restrições de arestas estão sendo obedecidas

$$\sum_{i \in S} x_{ij} \leq w_j, \quad \forall \text{ clique } S \subseteq V, \quad \forall 1 \leq j \leq n$$

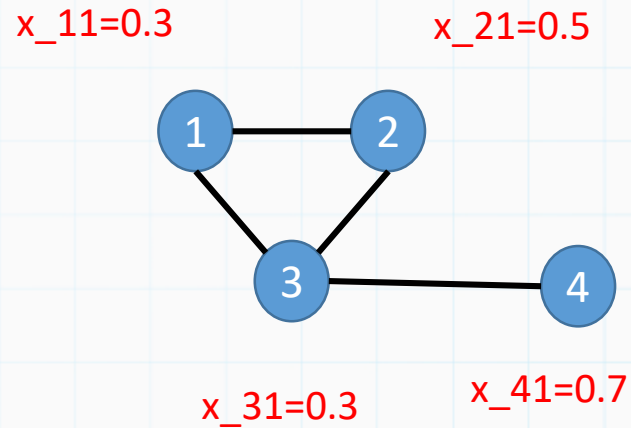
Tarefa de Casa



Ex:

- Na função de “usercut” do “callback”, vamos pegar as variáveis. Vamos montar o grafo residual da cor j.

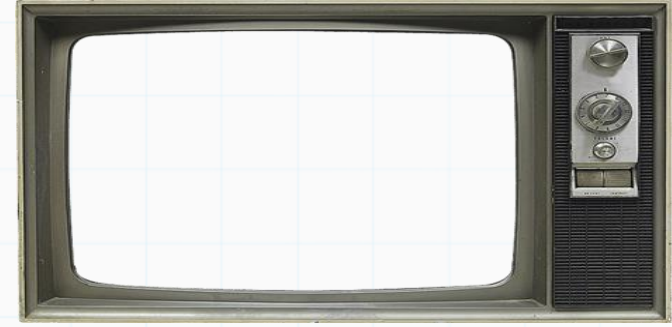
$w_1=1$



veja que a clique $S=\{1,2,3\}$
está violada por 0.1

$$\sum_{i \in S} x_{ij} \leq w_j, \quad \forall \text{ clique } S \subseteq V, \quad \forall 1 \leq j \leq n$$

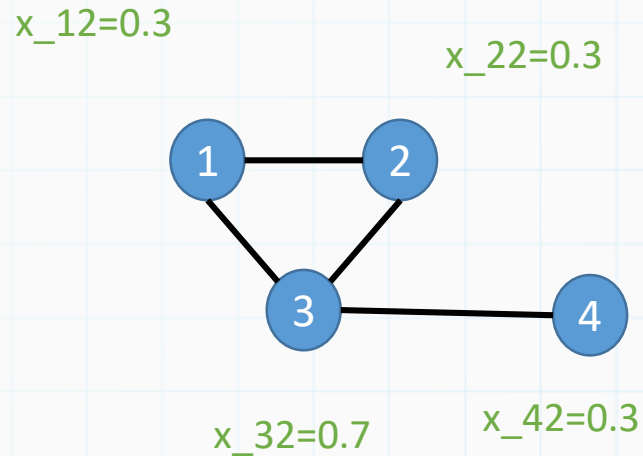
Tarefa de Casa



Ex:

- Na função de “usercut” do “callback”, vamos pegar as variáveis. Vamos montar o grafo residual da cor j.

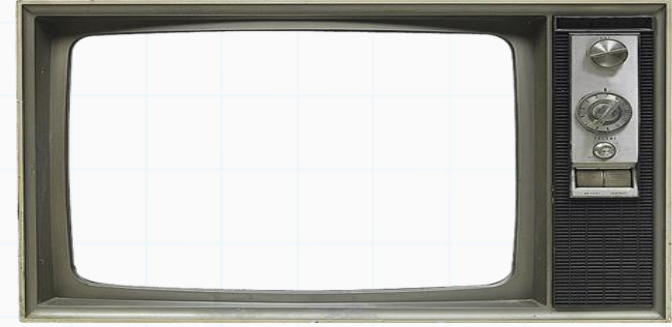
$w_2=1$



veja que a clique $S=\{1,2,3\}$
está violada por 0.3

$$\sum_{i \in S} x_{ij} \leq w_j, \quad \forall \text{ clique } S \subseteq V, \quad \forall 1 \leq j \leq n$$

Tarefa de Casa



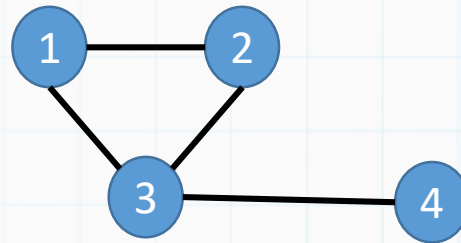
Ex:

- Na função de “usercut” do “callback”, vamos pegar as variáveis. Vamos montar o grafo residual da cor j.

$$w_3=0.5$$

$$x_{13}=0.4$$

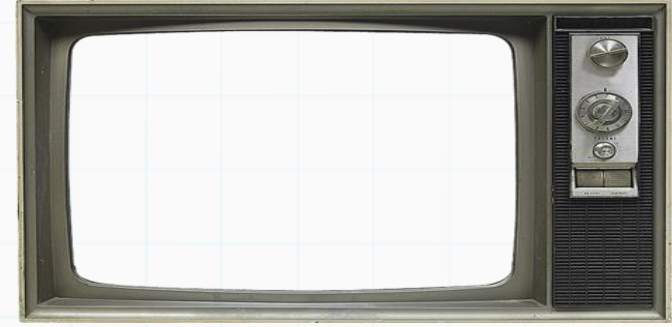
$$x_{23}=0.2$$



Sem cliques violadas

$$\sum_{i \in S} x_{ij} \leq w_j, \quad \forall \text{ clique } S \subseteq V, \quad \forall 1 \leq j \leq n$$

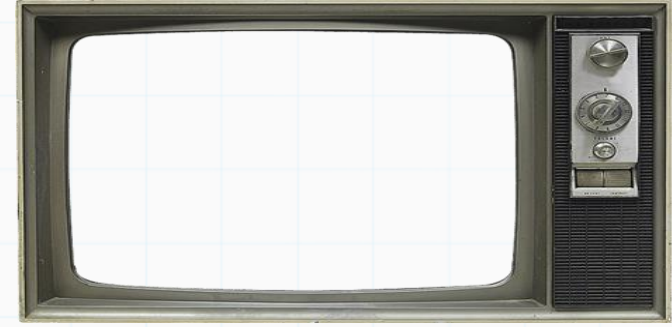
Tarefa de Casa



Algoritmo:

- Para cada cor $j=1\dots n$ faça
 - monte grafo residual G_j
 - enquanto **<critério de parada>**
 - encontrar clique S violada (**heurística**) e inserir no modelo

Tarefa de Casa



Algoritmo:

- Para cada cor $j=1\dots n$ faça
 - monte grafo residual G_j
 - enquanto **<critério de parada>**
 - encontrar clique S violada (**heurística**) e inserir no modelo

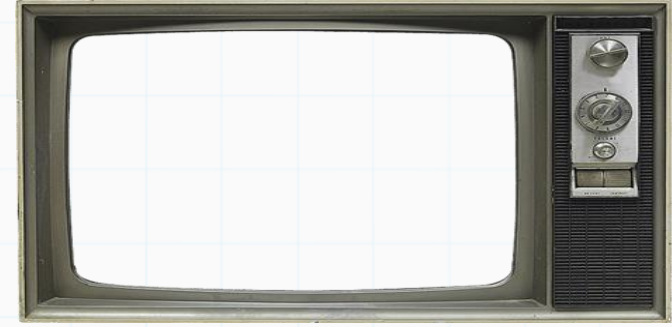
Heurística(G_j):

ex: ordenar vértices em relação ao valor da cor j .
percorrer vértices na ordem e procurando montar
a clique máxima.

add-nos:

- multi-start aleatório
- buscas locais
- o céu é o limite...

Tarefa de Casa



Algoritmo:

- Para cada cor $j=1\dots n$ faça
 - monte grafo residual G_j
 - enquanto **<critério de parada>**
 - encontrar clique S violada (**heurística**) e inserir no modelo

Heurística(G_j):

ex: ordenar vértices em relação ao valor da cor j .
percorrer vértices na ordem e procurando montar a clique máxima.

add-nos:

- multi-start aleatório
- buscas locais
- o céu é o limite...

Critério de parada:

- número fixo de cortes
- cortes violados por um determinado valor ϵ
- tempo
- ...

Tente achar no
mínimo um corte
clique por cor j

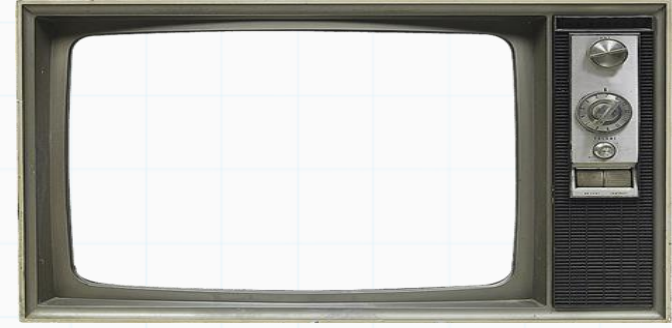
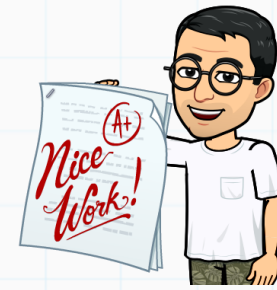
Tarefa de Casa

- resolver as 3 instancias fornecidas:

TPI_BC_COL_0.txt (ótimo é 5)

TPI_BC_COL_1.txt

TPI_BC_COL_2.txt



Formato do arquivo:

p edge n (numero de vertices) m (numero de arestas)
e vertice(i) vertice(j) (lista de arestas)

TPI_BC_COL_0.txt

EX:

p edge 25 320

e 1 7

e 1 13

e 1 19

e 1 25

e 1 2

e 1 3

e 1 4

e 1 5

e 1 6

e 1 11

e 1 16

...

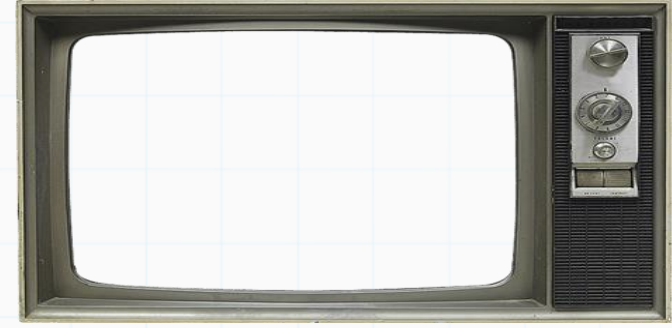
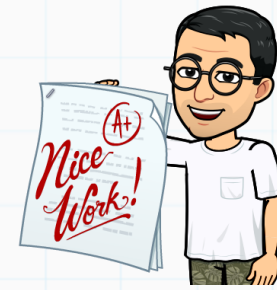
Tarefa de Casa

- resolver as 3 instancias fornecidas:

TPI_BC_COL_0.txt (ótimo é 5)

TPI_BC_COL_1.txt

TPI_BC_COL_2.txt



Formato do arquivo:

p edge n (numero de vertices) m (numero de arestas)
e vertice(i) vertice(j) (lista de arestas)

O programa deverá dar ao usuário a opção de usar os cortes cliques (Branch-and-cut) ou executar sem corte nenhum (Branch-and-bound). Além disso, deverá imprimir como saída:

- 1) o valor da solução ótima,
- 2) tempo total,
- 3) número de cortes gerados no total e
- 4) número de nós analisados na árvore de busca.

TPI_BC_COL_0.txt

EX:

p edge 25 320

e 1 7

e 1 13

e 1 19

e 1 25

e 1 2

e 1 3

e 1 4

e 1 5

e 1 6

e 1 11

e 1 16

...

Até a próxima

