

# Contents

<b>1</b>	<b>Matchings</b>	<b>2</b>
1.1	Como interpretar matrices de adjacencia? . . . . .	2
1.1.1	Que representan? . . . . .	2
1.1.1.1	A quien estan conectados los vertices de X?	2
1.1.1.2	A quien estan conectados los vertices de Y?	2
1.1.1.3	Que capacidad tienen los lados?	2
1.1.2	Como es la estructura de la matriz? . . . . .	3
1.2	Como se arman los caminos usando solo la matriz? . . . . .	3
1.2.1	Algoritmo . . . . .	3
1.2.1.1	Tips	3
1.2.1.2	Ejemplo	4
<b>3</b>	<b>Min-max</b>	<b>5</b>
3.1	Pasos . . . . .	5

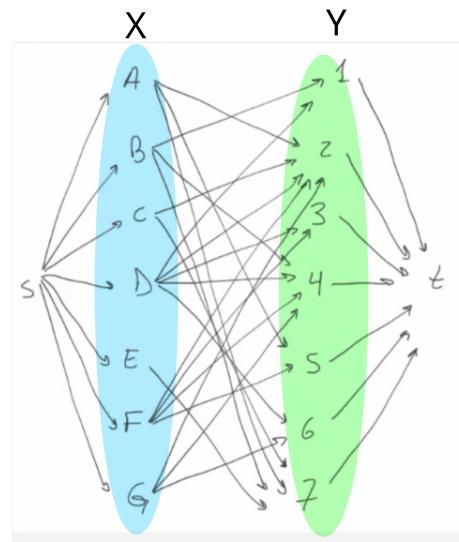
# 1 Matchings

## 1.1 Como interpretar matrices de adjacencia?

### 1.1.1 Que representan?

Un network, cuyos vertices pertenecen a uno de dos conjuntos, llamemoslos X e Y.  
(s y t son la excepcion)

	1	2	3	4	5	6	7
A	0	1	0	0	1	0	1
B	1	0	0	1	0	0	1
C	0	1	0	0	0	0	1
D	1	1	1	1	0	1	0
E	0	0	0	0	0	0	1
F	0	1	1	1	1	0	0
G	0	1	0	1	0	1	0



#### 1.1.1.1 A quien estan conectados los vertices de X?

Estan conectados a s

#### 1.1.1.2 A quien estan conectados los vertices de Y?

Estan conectados a t

#### 1.1.1.3 Que capacidad tienen los lados?

Siempre tienen capacidad 1

### **1.1.2 Como es la estructura de la matriz?**

Las filas son nodos que perteneces a X y las columnas nodos que pertenecen a Y.

### **1.2 Como se arman los caminos usando solo la matriz?**

Se buscan los 1s en cada fila, ya que sabemos que los vertices de las filas estan conectados a s y los de las columnas estan conectados a t

#### **1.2.1 Algoritmo**

Por cada fila de la matriz:

1. Marcar el primer 1 de la fila que no posea otro 1 marcado en la misma columna
2. Si todas los 1s disponibles estan en una columna ocupada:
  1. Marcar la columna bloqueada con el nombre del vertice de la fila
  2. Marcar fila que bloquedo la columna con el nodo de la columna
  3. Ver si el nodo que bloquedo la columna puede mandar flujo por otro lado
  4. Repetir (1), (2) y (3) hasta que se encuentre una forma de mandar el flujo
  5. Si no se encuentra: no hay match?
3. Seguir con la siguiente fila
  - Si la fila no llega a t, marcar fila con un (\*)

##### **1.2.1.1 Tips**

- Al encontrarse con una columna bloqueada pensar el proceso como un camino backward de EK o Dinic
- Las filas o columnas se marcan con los nodos que los agregaron al camino resultante de hacer EK.

##### **1.2.1.2 Ejemplo**

	1	2	3	4	5	6	7
A	0	1	0	0	1	0	1
B	1	0	0	1	0	0	1
C	0	1	0	0	0	0	1
D	1	1	1	1	0	1	0
E	0	0	0	0	0	1	
F	0	1	1	1	0	0	
G	0	1	0	1	0	1	0

*C*                    *E*

	1	2	3	4	5	6	7
A	0	1	0	0	1	0	1
B	1	0	0	1	0	0	1
C	0	1	0	0	0	0	1
D	1	1	1	1	0	1	0
E	0	0	0	0	0	0	1
F	0	1	1	1	0	1	0
G	0	1	0	1	0	1	0

*f)*                    *f)*

# 2

## 3 Min-max

### 3.1 Pasos

1. Armar conjunto de posibles valores de la matriz
2. Buscar el medio del conjunto (busqueda binaria), llamemoslo  $M$
3. Armar matriz con valores  $\leq M \rightarrow 1$  y  $> M \rightarrow 0$
4. Ver si hay match:
  - a. Si hay: Hacer busqueda binaria con  $M$  como limite superior
  - b. Si no hay: Hacer busqueda binaria con  $M$  como limite inferior