

# Flujo bis

Julián Braier

Departamento de Computación - FCEyN UBA

2024 1c

# Escenas del Capítulo Anterior



## Sin conexión a Internet

Prueba a:

- Comprobar los cables de red, el módem y el router
- Volver a conectarte a una red Wi-Fi
- [Ejecutar Diagnósticos de red de Windows](#)

ERR\_INTERNET\_DISCONNECTED

## Code ▲



```
1 #include<bits/stdc++.h>
2
3 using namespace std;
4
5 int main(){
6
7     return 0;
8 }
```



[SHARE CODE TO OTHERS](#)

## Test details ▲

### Test 1

Verdict: **WRONG ANSWER**

input	
2 1	 
1 2	

correct output	
1	 
1 2	

user output	
(empty)	

## Submission details

Task:	<a href="#">Police Chase</a>
Sender:	Agua_Podrida
Submission time:	2024-06-19 18:18:29 +0300
Language:	C++20
Status:	COMPILE ERROR

## Compiler report ▲

```
input/code.cpp:16:30: error: expected ',' or '...' before numeric constant
16 |         void agregar_eje(int 1u,
   |                             ^~
input/code.cpp:75:14: error: expected ')' before ';' token
75 |         int n;
   |         ^
   |         )
input/code.cpp:16:25: note: to match this '('
16 |         void agregar_eje(int 1u,
   |                             ^
input/code.cpp: In constructor 'grafo::grafo(int)':
input/code.cpp:10:17: error: 'n' was not declared in this scope; did you mean 'yn'?
10 |         n = _n;
   |         ^
   |         yn
input/code.cpp: In function 'int main()':
input/code.cpp:88:30: error: no matching function for call to
'grafo::agregar_eje(int&, int&, int)'
88 |         g.agregar_eje(u,v,1);
   |         ~~~~~~^~~~~~
input/code.cpp:16:14: note: candidate: 'void grafo::agregar_eje(int)'
16 |         void agregar_eje(int 1u,
   |         ~~~~~~^...

```

## Submission details

Task:	<a href="#">Police Chase</a>
Sender:	Agua_Podrida
Submission time:	2024-06-19 18:20:22 +0300
Language:	C++20
Status:	COMPILE ERROR

## Compiler report ▲

```
input/code.cpp:16:30: error: expected ',' or '...' before numeric constant
   16 |         void agregar_eje(int 1u,
      |                             ^~
input/code.cpp:75:14: error: expected ')' before ';' token
   75 |         int n;
      |         ^
      |     )
input/code.cpp:16:25: note: to match this '('
   16 |         void agregar_eje(int 1u,
      |                             ^
input/code.cpp: In constructor 'grafo::grafo(int)':
input/code.cpp:10:17: error: 'n' was not declared in this scope; did you mean 'yn'?
   10 |         n = _n;
      |         ^
      |     yn
input/code.cpp: In function 'int main()':
input/code.cpp:88:30: error: no matching function for call to
'grafo::agregar_eje(int&, int&, int)'
   88 |         g.agregar_eje(u,v,1);
      |         ~~~~~~^~~~~~
input/code.cpp:16:14: note: candidate: 'void grafo::agregar_eje(int)'
   16 |         void agregar_eje(int 1u,
      |         ~~~~~~^..
```

## Submission details

Task:	<a href="#">Police Chase</a>
Sender:	Agua_Podrida
Submission time:	2024-06-19 18:20:48 +0300
Language:	C++20
Status:	COMPILE ERROR



## Compiler report ▲



```
input/code.cpp: In function 'int main()':
input/code.cpp:96:20: error: 'padre' was not declared in this scope
   96 |         if(padre[v] != -1){
      |         ~~~~~^
```



## Test details ▲

### Test 1

Verdict: **WRONG ANSWER**

input	
2 1	 
1 2	

correct output	
1	 
1 2	

user output	
1	 
2 1	
2 1	

```
void agregar_eje(int u, int v, int c){  
    ady[u].push_back(v);  
    ady[v].push_back(u);  
    capacidad[u][v] = c;  
}
```



## Submission details

Task:	<a href="#">Police Chase</a>
Sender:	Agua_Podrida
Submission time:	2024-06-19 18:23:24 +0300
Language:	C++20
Status:	COMPILE ERROR

## Compiler report ▲

```
input/code.cpp: In function 'int main()':
input/code.cpp:98:57: error: 'capacidad' was not declared in this scope
   98 |                                     if(g.padre[u] == -1 and capacidad[v][u]){
      |                                     ^~~~~~
```

```

//padre[v] = -1 sii v no es alcanzable desde s
for(int v = 1; v <= n; v++){
    if(g.padre[v] != -1){
        for(auto u : g.ady[v]){
            if(g.padre[u] == -1){
                cout<<u<<' '<<v<<'\n';
            }
        }
    }
}

```

Figure: Versión que dio WA. La versión que vimos en clase (excepto que esta sí compila).

```

//padre[v] = -1 sii v no es alcanzable desde s
for(int v = 1; v <= n; v++){
    if(g.padre[v] != -1){
        for(int u = 1; u <= n; u++){
            if(g.padre[u] == -1 and g.capacidad[v][u]){
                cout<<v<<' '<<u<<'\n';
            }
        }
    }
}

```

Figure: Versión que dio AC.

# Recursos de Modelado

- Múltiples fuentes  $s_1, \dots, s_p$  y sumideros  $t_1, \dots, t_q$ .

# Recursos de Modelado

- Múltiples fuentes  $s_1, \dots, s_p$  y sumideros  $t_1, \dots, t_q$ .
  - Reducimos agregando fuente  $s$  y sumidero  $t$  ficticios, con los ejes  $(s, s_i)$  y  $(t_j, t)$  con capacidad infinito.

# Recursos de Modelado

- Múltiples fuentes  $s_1, \dots, s_p$  y sumideros  $t_1, \dots, t_q$ .
  - Reducimos agregando fuente  $s$  y sumidero  $t$  ficticios, con los ejes  $(s, s_i)$  y  $(t_j, t)$  con capacidad infinito.
- Capacidad en los nodos.

# Recursos de Modelado

- Múltiples fuentes  $s_1, \dots, s_p$  y sumideros  $t_1, \dots, t_q$ .
  - Reducimos agregando fuente  $s$  y sumidero  $t$  ficticios, con los ejes  $(s, s_i)$  y  $(t_j, t)$  con capacidad infinito.
- Capacidad en los nodos.
  - Partimos el nodo en dos, y los conectamos con un eje.

# Recursos de Modelado

- Múltiples fuentes  $s_1, \dots, s_p$  y sumideros  $t_1, \dots, t_q$ .
  - Reducimos agregando fuente  $s$  y sumidero  $t$  ficticios, con los ejes  $(s, s_i)$  y  $(t_j, t)$  con capacidad infinito.
- Capacidad en los nodos.
  - Partimos el nodo en dos, y los conectamos con un eje.
- Aristas bidireccionales.
  - Ponemos ejes en ambas direcciones.

# ¿Saben qué día es hoy?



# ¿Saben qué día es hoy?

Lionel Messi / Fecha de nacimiento

## 24 de junio de 1987





edad ~~36~~ años

37



# Copa América 2024

## Grupo A

Club	PJ	G	E	P	GF	GC	DG	Pts	Últimos 5
1  Argentina	1	1	0	0	2	0	2	3	✓ ○ ○ ○ ○
2  Chile	1	0	1	0	0	0	0	1	− ○ ○ ○ ○
3  Perú	1	0	1	0	0	0	0	1	− ○ ○ ○ ○
4  Canadá	1	0	0	1	0	2	-2	0	✗ ○ ○ ○ ○

# Mundial 1990

## Reglas [\[ editar \]](#)

Los 24 equipos que participan en la fase final se dividen en seis grupos de cuatro equipos cada uno. Dentro de cada grupo se enfrentan una vez entre sí, por el [sistema de todos contra todos](#). Según el resultado de cada partido se otorga dos puntos al ganador, un punto a cada equipo en caso de empate, y ninguno al perdedor.

### Grupo B [\[ editar \]](#)

Selección	Pts.	PJ	PG	PE	PP	GF	GC	Dif.
 Camerún	4	3	2	0	1	3	5	-2
 Rumania	3	3	1	1	1	4	3	1
 Argentina	3	3	1	1	1	3	2	1
 Unión Soviética	2	3	1	0	2	4	4	0

# Mundial 1990

## Reglas [\[ editar \]](#)

Los 24 equipos que participan en la fase final se dividen en seis grupos de cuatro equipos cada uno. Dentro de cada grupo se enfrentan una vez entre sí, por el [sistema de todos contra todos](#). Según el resultado de cada partido se otorga dos puntos al ganador, un punto a cada equipo en caso de empate, y ninguno al perdedor.

### Grupo B [\[ editar \]](#)

Selección	Pts.	PJ	PG	PE	PP	GF	GC	Dif.
 Camerún	4	3	2	0	1	3	5	-2
 Rumania	3	3	1	1	1	4	3	1
 Argentina	3	3	1	1	1	3	2	1
 Unión Soviética	2	3	1	0	2	4	4	0

## 24 de junio – Turín

 Brasil	0
 Argentina	1

# Problema I: Campeonato

Se juega un torneo de  $N$  equipos, en el que juegan todos contra todos una vez. Se asignan 2 puntos por ganar, 1 por empatar, ninguno por perder. Al final del torneo, gana el que sumó más puntos. En caso de empate, se juegan partidos entre los equipos empatados.

**Problema:** dados los resultados parciales de un torneo al que le quedan partidos pendientes y un equipo  $e$ . ¿Tiene  $e$  chance de ser campeón?

# Problemas que se pueden resolver con flujo máximo

En grafos bipartitos, hay varios problemas que se pueden resolver con flujo máximo.

- Matching máximo.
- Cubrimiento mínimo de vértices por aristas.
- Cubrimiento mínimo de aristas por vértices.
- Conjunto independiente máximo.

# Problemas que se pueden resolver con flujo máximo

- Matching máximo.
  - Un matching es un conjunto de aristas que no contiene un par de aristas incidentes al mismo vértice.
- Cubrimiento mínimo de vértices por aristas.
  - Un cubrimiento de vértices por aristas es un conjunto de aristas tal que todo vértice es incidente a al menos una arista en él.

# Problemas que se pueden resolver con flujo máximo

- Cubrimiento mínimo de aristas por vértices.
  - Un cubrimiento de aristas por vértices es un conjunto de vértices tal que toda arista es incidente a al menos un vértice en él.
- Conjunto independiente máximo.
  - Un conjunto independiente es un conjunto de vértices tal que no hay ninguna arista entre ellos.



# Si vas a leer bibliografía...

- Cubrimiento de aristas por vértices = cubrimiento de aristas = cubrimiento de vértices = vertex cover
- Cubrimiento de vértices por aristas = cubrimiento de vértices = cubrimiento de aristas = edge cover

## Problema II: Tortas<sup>1</sup>

La pastelería está planeando qué tortas preparar. Hay  $C$  recetas de torta, va a preparar a lo sumo una torta de cada tipo, supuestamente, la gente está dispuesta a pagar mucho más por su torta si esta es única. Se estima que la ganancia que se puede obtener al una torta del  $i$ -ésimo tipo es  $c_i$ . Cada receta necesita ciertas herramientas para su preparación, estas pueden ser reutilizadas para preparar distintas tortas. Hay  $T$  herramientas,  $t_j$  es el costo de la  $j$ -ésima. Inicialmente, la pastelería no tiene ninguna herramienta.

**Problema:** Dadas:  $c_1, \dots, c_C$ , las ganancias por cada tipo de torta,  $t_1, \dots, t_T$ , los costos de cada herramienta y una matriz  $B$  de booleanos tal que  $b_{i,j}$  sii la herramienta  $j$  es necesaria para la receta  $i$ , computar la máxima ganancia que se puede obtener.

**Límites:**  $1 \leq C, T \leq 200$  y  $0 \leq c_i, t_j \leq 10^9$ . 4

---

<sup>1</sup>Tomado (pero simplificado) del Regional 2023 ICPC de Europa Central  
<https://cerc2023.acm.si/>.

# Problema III: Lamponne y las torres<sup>2</sup>

## Ejercicio 3: Torres

En una cómoda tarde de primavera, Lamponne y Santos se encontraban jugando no tan apaciblemente al ajedrez. Tras su tercer derrota, Lamponne arrojó el tablero de mármol por los aires, logrando como resultando que tras la caída varios casilleros del mismo se resquebrajaran.

Santos, en vez de enojarse, observó con curiosidad que ahora, en este tablero con casilleros “tapados”, era posible colocar más de una torre por fila sin que estas se amenazen, si se interpreta que dos torres se amenazan mutuamente si y solamente si se encuentran en la misma fila (o columna) y no hay un casillero roto entre ellos. Interesado en esta cualidad, se propuso encontrar la máxima cantidad de torres que se pueden colocar en este tablero sin que se amenazen mutuamente.

Lamponne, que sigue herido por las tres derrotas, nos pide ayuda para calcular este número antes que Santos.

---

<sup>2</sup>Propuesto por Santi Cifu para el TP3 del cuatri pasado.

# Problema III: Lamponne y las torres

## Descripción de una instancia

La entrada contiene múltiples casos de test. La primera línea consiste de un número  $c$  indicando su cantidad.

La primera línea de cada caso contiene un entero  $N$  ( $1 \leq N \leq 100$ ), indicando la cantidad de filas y columnas del tablero. Luego siguen  $N$  líneas con  $N$  enteros cada una, describiendo el mismo. Si el  $j$ -ésimo carácter de la  $i$ -ésima fila es un 1, entonces ese casillero (el de la fila  $i$  y columna  $j$ ) está roto, mientras que caso contrario, si el entero es un 0, entonces está intacto.

## Descripción de la salida

Para cada caso de test se debe imprimir una línea indicando la máxima cantidad de torres que pueden colocarse en este tablero resquebrajado siguiendo las reglas que dispuso Santos.

**Ejemplo de entrada y salida** Se presenta un ejemplo de entrada y su correspondiente salida.

Entrada de ejemplo	Salida esperada de ejemplo
3	2
2	4
0 0	3
0 1	
3	
0 0 0	
0 1 0	
0 0 0	
3	
1 0 0	
1 1 0	
0 0 1	

## Problema III: Lamponne y las torres

Las formas de colocar las torres en el segundo y el tercer caso de test se pueden ver a continuación:

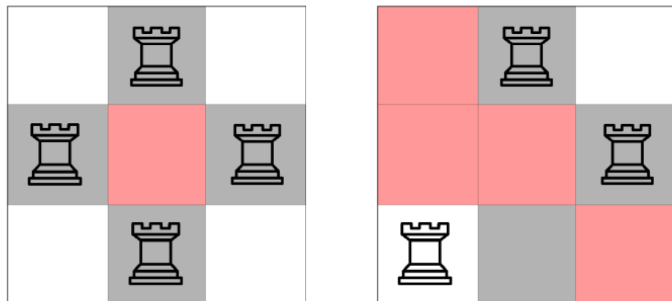


Figura 3: Formas de colocar la máxima cantidad de torres posibles para el segundo y tercer caso de test.

Disculpe

# Disculpe

¿Fuego tiene?