

LA MÁQUINA CALCULADORA

Ejercicio del Juez de la máquina calculadora.



U N I V E R S I D A D
COMPLUTENSE
M A D R I D

ALBERTO VERDEJO

La máquina calculadora

- ▶ Javier le ha construido a su hijo Luis una máquina para calcular.



Máquina con un display de 4 dígitos



Botones para modificar el valor que aparece en el display
Las operaciones se utilizan con el módulo 10000.

Si hacemos $5000 * 2$ el resultado que aparecerá es 0

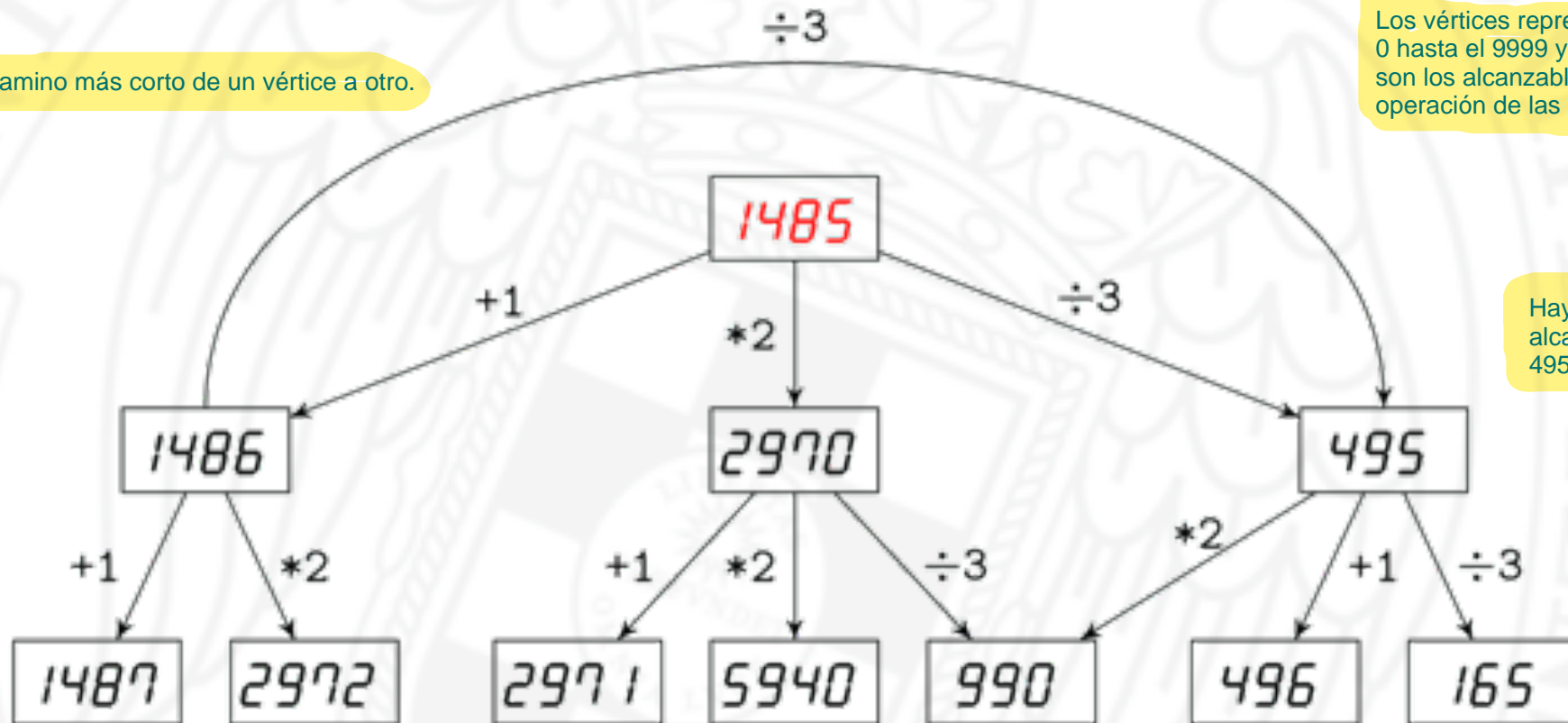
- ▶ Javier configura la máquina con un número y reta a Luis a que consiga otro número dado pulsando los botones el menor número de veces.

Modelado

Es un grafo fuertemente conexo.

- Podemos modelar el problema con un grafo dirigido donde los vértices son los números y los adyacentes son los números obtenibles con cualquiera de las tres operaciones.

Tenemos que hallar el camino más corto de un vértice a otro.



Los vértices representan los números desde el 0 hasta el 9999 y los adyacentes a cada número son los alcanzables al hacer cualquier operación de las posibles.

Hay vértices que se pueden alcanzar desde varios vértices. 495 desde 1486 y desde 1485

Solución

- ▶ Realizamos un **recorrido en anchura** en ese grafo comenzado por el número origen.
- ▶ El grafo es **implícito**, puede irse generando a la vez que se recorre.
- ▶ Podemos parar el recorrido en cuanto lleguemos al número destino.
- ▶ Implementación *ad hoc*.

No hace falta crearlo desde el principio.

En la mayoría de casos construir todo el grafo desde el principio sería hacer trabajo de más.

Implementación

Módulo de las operaciones.

```
const int MAX = 10000;  
const int INF = 1000000000; // ∞  
  
int adyacente(int v, int i) {  
    switch (i) {  
        case 0: return (v + 1) % MAX; // + 1  
        case 1: return (v * 2) % MAX; // * 2  
        case 2: return v / 3;          // / 3  
    }  
}
```


Implementación

```
int bfs(int origen, int destino) {  
    if (origen == destino) return 0;  
    vector<int> distancia(MAX, INF);  
    distancia[origen] = 0;  
    queue<int> cola; cola.push(origen);  
    while (!cola.empty()) {  
        int v = cola.front(); cola.pop();  
        for (int i = 0; i < 3; ++i) {  
            int w = adyacente(v,i);  
            if (distancia[w] == INF) {  
                distancia[w] = distancia[v] + 1;  
                if (w == destino) return distancia[w];  
                else cola.push(w);  
            }  
        }  
    }  
}
```

Si el origen y el destino coinciden, la distancia es 0.

Vector de distancias. Si la distancia al vértice es infinito es que TODAVÍA NO HA SIDO VISITADO.

Empieza con distancia 0-

Recorremos sus tres adyacentes.

$O(10000+30000)$

Siendo 10000 los vértices y 30000 las aristas.

Implementación

```
bool resuelveCaso() {  
    int origen, destino;  
    cin >> origen >> destino;  
  
    if (!cin) return false;  
  
    cout << bfs(origen, destino) << "\n";  
  
    return true;  
}
```