# Generar y probar: Combinaciones y el problema del clique

Teoría de Algoritmos I (75.29 / 95.06)

Ing. Víctor Daniel Podberezski
vpodberezski@fi.uba.ar

# Generar y probar: Combinaciones

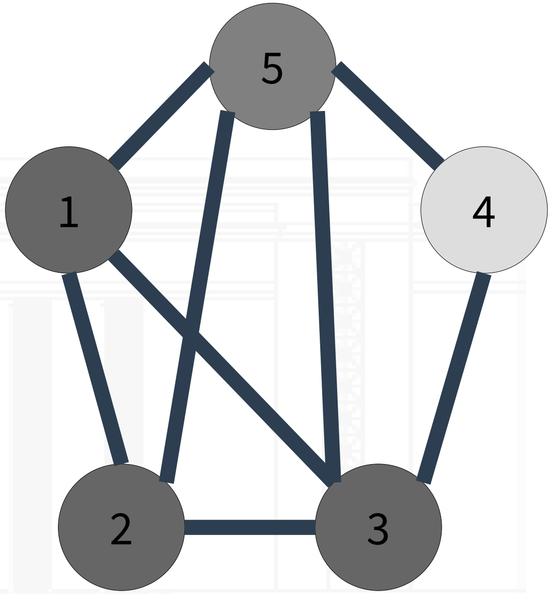
- **Espacio de soluciones**
  - Trabajaremos sobre la selección de un subconjunto de m elementos entre n.
  - Los elementos son únicos e indivisibles.
  - No importa el orden en el que son seleccionados

- **Función generativa**
  - Generación de las combinaciones de los elementos
  - Corresponderá a las restricciones explicitas del problema

# Ejemplo práctico: Problema del clique

- **Dado un grafo no direccionado G=(V,E)**
  - con V su conjunto de vértices y E el de sus aristas.
  - Queremos obtener, si existe, un clique de tamaño m dentro de él.

- **Un clique es un subconjunto de vértices**
  - en el que para cualquier par de ellos existe un eje que los une
  - Equivale a decir: todos los vértices son adyacentes entre sí
  - Equivale a decir: conforman entre sí un subgrafo completo




Clique de tamaño 4: {1,2,3,5}

# Ejemplo práctico: Problema del clique

**Para un subconjunto M vértices de tamaño m del grafo G=(V,E)**

Podemos verificar si cada vértices este conectado entre sí

Es un proceso en $O(m^2)$.

**Si todos los vértices de M están conectados entre si**

Corresponde a un clique de tamaño m

**La cantidad de subconjuntos M de tamaño m en G**

Se puede calcular como $\binom{n}{m} = \frac{n!}{m!(n-m)!}$

# Ejemplo práctico: Problema del clique

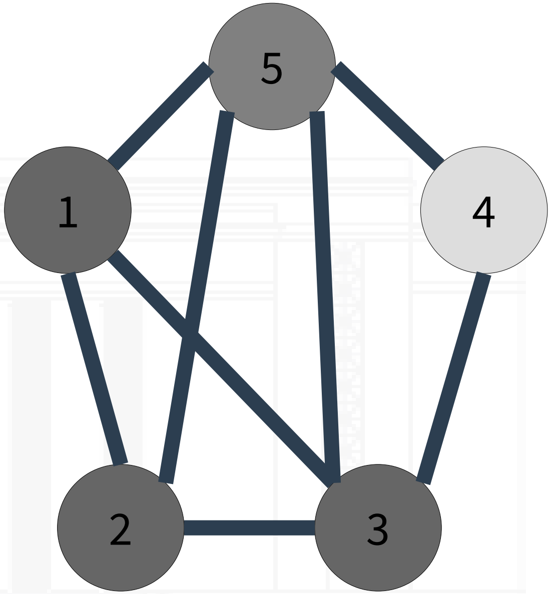
## Cada vértice

Recibe un identificador numérico entre 1 y n.

## Para un subconjunto M de vértices de G

Utilizaremos un vector C de tamaño m

Los vértices en el vector estarán ordenados de forma decreciente




C:

| | | | |
|---|---|---|---|
| 5 | 3 | 2 | 1 |

# Generación de las combinaciones de n elementos tomados de a m

- **Se generarán en orden lexicográfico.**
  - Se comenzará los m elementos con menor identificador
- **Generar la próxima combinación**

  Utilizará dos variables centinelas en el vector

  Se buscara desde la derecha el primer elemento no contiguo y se lo incrementa
- **Finaliza el proceso**
  - cuando el vector contenga los m elementos con mayor identificador
  - El primer elemento no contiguo es el primer centinela

$\underbrace{\text{0 \quad 8}}_{\text{centinelas}}$ Con n=7 y m=4

| | | | | | |
|---|---|---|---|---|---|
| 0 | 8 | 4 | 3 | 2 | 1 |
| 0 | 8 | 5 | 3 | 2 | 1 |
| 0 | 8 | 5 | 4 | 2 | 1 |
| 0 | 8 | 5 | 4 | 3 | 1 |
| 0 | 8 | 5 | 4 | 3 | 2 |
| 0 | 8 | 6 | 3 | 2 | 1 |
| 0 | 8 | 6 | 4 | 2 | 1 |
| … | | … | | | |
| 0 | 8 | 7 | 6 | 5 | 4 |

n+1

# Generación de las combinaciones: Inicialización

## Para inicializar:

Establecer en cada posición C[i] el elemento i ordenado lexicográficamente

Establecer el primer centinela con valor n+1

Establecer el segundo centinela con valor 0

Complejidad temporal: O(m)

| | | | | | |
|---|---|---|---|---|---|
| 0 | 8 | 4 | 3 | 2 | 1 |
| 6 | 5 | 4 | 3 | 2 | 1 |

```
Sea C un vector representando el conjunto de
vértices a evaluar con dos posiciones
adicionales.

Inicializar C:
     Desde i=1 a m
          C[i]=i
     C[k+1]=n+1
     C[k+2]=0
```

# Generación de las combinaciones: Obtener siguiente

## Buscar el primer elemento desde la derecha no contiguo

Incrementar en 1 ese elemento

Establecer los elementos anteriores con valor contiguos comenzando por el uno.

| | | | | | |
|---|---|---|---|---|---|
| 0 | 8 | 5 | 4 | 3 | 2 |
| 6 | 5 | 4 | 3 | 2 | 1 |

| | | | | | |
|---|---|---|---|---|---|
| 0 | 8 | 6 | 3 | 2 | 1 |

```
Incrementar C:
    Sea j=1
    Mientras C[j]+1==C[j+1]
        C[j] =  j
        j+=1
    Si j>m
        retornar 'fin'

    C[j]+=1
    retornar C
```

# Generación de las combinaciones: Finalización

## El proceso finaliza

Cuando todos los elementos son contiguos excepto el último en el vector

| | | | | | |
|---|---|---|---|---|---|
| 0 | 8 | 7 | 6 | 5 | 4 |
| 6 | 5 | 4 | 3 | 2 | 1 |

```
Incrementar C:
    Sea j=1
    Mientras C[j]+1==C[j+1]
        C[j] =  j
        j+=1
    Si j>m
        retornar 'fin'

    C[j]+=1
    retornar C
```

# Verificación de la solución

## Verificación:

Evaluar si los m elementos corresponden a vértices unidos entre sí.

Complejidad temporal: $O(m^2)$

```
Sea C un vector con los nodos a evaluar
Sea M la matriz de adyacencia del grafo G

Es_clique C:
    Desde elemento1=1 a m
        Desde elemento2=elemento1+1 a m
            Si M[C[elemento1],C[elemento2]]==0
                Retornar ‘No’
    Retornar ‘Si’
```

# Generar y probar: Pseudocódigo

## Unificando:

Por cada posible subconjunto de m vértices verificamos si corresponde a un clique.

Si lo es, lo retornamos.

Sino, se continua buscando hasta verificar todos los posibles subconjuntos.

Complejidad temporal: $O(m^2 \; n! / m!(n-m)!)$

```
Sea C un vector con los nodos a evaluar
Sea M la matriz de adyacencia del grafo G

Inicializar C

Repetir
     Si Es_clique C
          retornar C
Hasta que Incrementar C == ‘fin’

retornar ‘No hay solución’
```