

# MM2015: Proyecto Códigos y Grupos

Para entregar el Martes, Dic 1, 2009

*H. Villafuerte*

**Carlos E. López Camey**

## Grupos.

### (a) Seleccionando un grupo finito:

Utilizemos  $S_n$

### (b) Seleccionemos un entero $n$ que determinará el orden del grupo elegido

```
sage: n = round(random()*10)
sage: n
9.0
sage: G = SymmetricGroup(9)
sage: G.order()
362880
```

Vemos que el orden de  $G$  es de 362880.

### (c) Obtengamos un listado $H$ de subgrupos de $G$

```
sage: H = G.normal_subgroups()
sage: H
[Permutation Group with generators [],
 Permutation Group with generators [(2,3)(4,5), (2,4,3), (2,4,6,5,3), (2,5,3)(4,7,6), (2,8,7,4,5,6),
 Permutation Group with generators [(1,2), (1,2,3,4,5,6,7,8,9)]]
```

### (d) Orden de los subgrupos $H$

```
sage: [subgrupo.order() for subgrupo in H]
[1, 181440, 362880]
```

Vemos que el último subgrupo del listado  $H$  tiene el mismo orden que el grupo inicial  $G$ .

```
sage: G.order() == H[2].order()
True
```

### (e) Repitamos algunas veces

Hagamos lo siguiente en sage

```
Grupo = SymmetricGroup(7);

def verificar(G,H):
    return G.order() == H[len(H)-1].order()

for n in range(1,20):
    G = SymmetricGroup(n)
    print verificar(G,G.normal_subgroups())
```

Vemos como respuesta

```
True
True
True
True
True
True
True
True
True
True
True
True
True
True
True
True
True
True
True
True
```

Lo cual, demuestra nuestra congetura para todos los grupos  $S_1 \dots S_{20}$  aunque no para todos.

## Códigos

### (a) Escojamos un mensaje $M$

Escojamos "Superfragilisticoespiralidoso"

```
M = "Superfragilisticoespiralidoso"
```

### (b) Codifiquemos $M$ empleando Hamming(7,4) sobre un canal que introduce 1-bit de error

```
sage: C = HammingCode(3,GF(2))
sage: G = C.gen_mat()
sage: H = C.check_mat()
sage: Mv = str2vec(M,4)
sage: Mv = [c*G for c in Mv]
sage: Mv = [c + error_canal(7,1) for c in Mv]
```

```
sage: Mv = [C.decode(c) for c in Mv]
sage: Mv = [G.solve_left(c).list() for c in Mv]
sage: Mhat = vec2str(Mv)
sage: print M, '->', Mhat
Superfragilisticoespiralidoso -> Superfragilisticoespiralidoso
sage: print M==Mhat
True
```

Vemos que el mensaje  $M$  se pudo recuperar.

**(c) Codifiquemoslo ahora sobre un canal con 2-bits de error**

```
sage: C = HammingCode(3,GF(2))
sage: G = C.gen_mat()
sage: H = C.check_mat()
sage: Mv = str2vec(M,4)
sage: Mv = [c*G for c in Mv]
sage: Mv = [c + error_canal(7,1) for c in Mv]
sage: Mv = [C.decode(c) for c in Mv]
sage: Mv = [G.solve_left(c).list() for c in Mv]
sage: Mhat = vec2str(Mv)
sage: print M==Mhat
False
```

Vemos que el mensaje  $M$  NO se pudo recuperar.