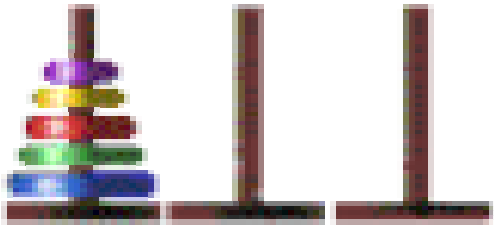
Esempio Torre di Hanoi

- Tre perni 1, 2, 3
- Pila di dischi dimensione crescente su perno 1
- vincoli:
  - uno solo disco alla volta può essere spostato
  - disco più grande mai sopra uno più piccolo



Problema: come spostare m dischi dal perno 1 al perno 2?

Algoritmo:
1. Spostare n-1 dischi da 1 a 3
2. Spostare m-esimo da 1 a 2
3. Spostare n-1 dischi da 3 a 2

Esempio Torre di Hanoi

Algoritmo:
1. Spostare n-1 dischi da 1 a 3
2. Spostare m-esimo da 1 a 2
3. Spostare n-1 dischi da 3 a 2

Si utilizza una funzione muovi:

```
void muovi(int n, int s, int d, int a)
{
 if (n == 1)
     muoviUnDisco(s, d);
 else {
     muovi(n-1, s, a, d);
     muoviUnDisco(s, d);
     muovi(n - 1, a, d, s);
 }
}  /* muovi */
```

Esempio Torre di Hanoi

```
int main(void)
{
  int dischi;    /* numero di dischi */
  int s, d;      /* pali sorgente e destinazione */

  printf("Numero di dischi? ");
  scanf("%d", &dischi);
  printf("Palo sorgente?     [1, 2 o 3] ");
  scanf("%d", &s);
  printf("Palo destinazione? [1, 2 o 3] ");
  scanf("%d", &d);

  muovi(dischi, s, d, 6 - d - s);
  return 0;

}  /* main */
```

Esempio Torre di Hanoi

Quando si utilizza la ricorsione multipla, il numero di attivazioni può essere esponenziale nella profondità della chiamata ricorsiva (cioè nell'altezza della pila di attivazione)

Esempio Torre Hanoi:

att(n) = numero attivazioni di `muovi(…)` per n dischi = numero spostamento dischi

$$att(n) = \begin{cases} 1, & n=1 \\ 1+2\ att(n-1), & n>1 \end{cases}$$

$$att(n) > 2^{n-1}$$