1) Se arrojan simultáneamente n dados, cada uno con k caras numeradas de 1 a k. Queremos calcular todas las maneras posibles de conseguir la suma total $s \in \mathbb{N}$ con una sola tirada. Consideramos que los dados son distinguibles, es decir que si $n = 2$ y $k = 2$, entonces existen 2 posibilidades que suman $s = 3$: 1 en el primer dado y 2 en el segundo y viceversa.

a) Definir en forma recursiva la función $f : \mathbb{N}^2 \to \mathbb{N}$ tal que $f(n, s)$ devuelve la cantidad de formas de conseguir la suma s con n dados de k caras.

b) Demostrar que f tiene la propiedad de superposición de subproblemas.

c) Definir un algoritmo *top-down* para calcular $f(n, s)$ indicando claramente las estructuras de datos utilizadas y la complejidad resultante.

d) Escribir el (pseudo-)código del algoritmo top-down resultante.

Aprobación: definir y justificar correctamente f, indicando cómo se computa $f(n, s)$ en tiempo $O(nk \min\{s, nk\})$.

$$f(i,s) = \begin{cases} 0 & \text{si } (i=0 \wedge s \neq 0) \vee s > ik \\ 1 & \text{si } i=0 \wedge s=0 \\ \sum_{d=1}^{k} f(i-1, s-d) & \text{si no} \end{cases}$$

El llamado $f(n,s)$ retorna la solución.
Un estado (i,s) modela que hay i dados por tirar, y que tenemos que sumar s entre todos los i dados.

Si $s > ik$ retornamos 0 porque no hay forma de sumar s aún tirando el valor máximo k con los i dados.

También retornamos 0 si no hay más dados para tirar $(i=0)$ y necesitamos sumar algún valor distinto de 0 $(s \neq 0)$.

Si $i=0$ y $s=0$ estamos en un caso base exitoso. Retornamos 1 pues en efecto sumamos 0 si tiramos 0 dados.

En el caso recursivo modelamos haber tirado cada uno de los k posibles valores con el i-ésimo dado.

#Estados: $O(ns)$
#Llamados: $O(k^n)$ Hay k llamados por cada uno de los n dados.
Calcular 1 estado: $O(k)$ Hay k sumas.

A priori calcular todos los estados sería $O(nsk)$. Pero como definimos la guarda $s > ik$, cuando $i=n$ en el primer llamado a la función, si $s > nk$ ya podemos retornar 0 sin calcular ningún otro estado.

Por lo tanto s está acotado por nk, y solo calculamos estados donde $s \leq nk$. Calcular todos los estados es $O(nk \min\{s, nk\})$.

Hay superposición de problemas cuando: $nk \min\{s, nk\} \ll k^n$.

```
M[1...n][1...min{s,nk}] ← ⊥
f(i, s):
    if (i = 0 ∧ s ≠ 0) ∨ s > ik:  return 0
    if i = 0 ∧ s = 0:  return 1
    if M[i][s] = ⊥:
        r ← 0
        for d ← 1 to k:
            r ← f(i-1, s-d)
        M[i][s] ← r
    return M[i][s]
```