Rod-Cutting Problem

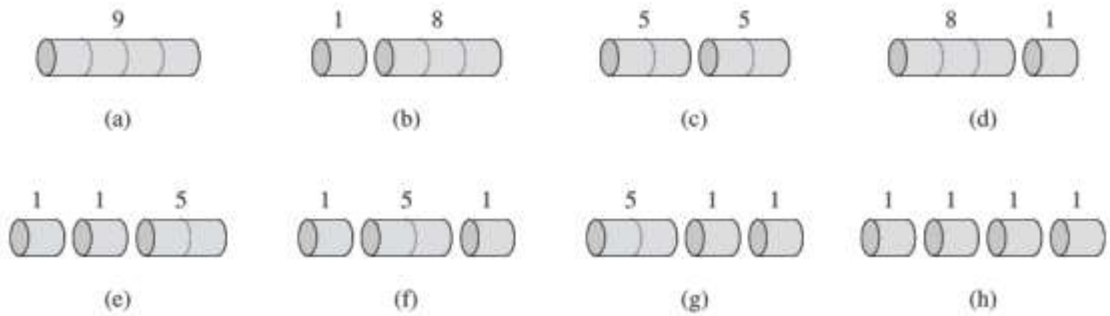
Se tiene una cuerda de n pulgadas y una tabla con los precios $p_i,\ i = 1,2,\dots,n$ según la longitud. Determinar la venta ***máxima*** r_n que se puede obtener al cortar la cuerda y vender los trozos.

Si el precio p_n es suficientemente elevado, no es necesario cortar la cuerda.

Ejemplo:

Longitud i	1	2	3	4	5	6	7	8	9	10
Precio p_i	1	5	8	9	10	17	17	20	24	30

Consideremos el caso cuando $n = 4$:



Estas son todas las formas posibles en las que una cuerda de 4 pulgadas puede ser recortada.

Nótese que la solución de la venta máxima es el literal c), cortar la cuerda en 2 trozos de 2 pulgadas.

Observe que podemos cortar una cuerda de longitud n en 2^{n-1} formas.

La solución óptima corta la cuerda en k trozos.

$$r_n = p_{i1} + p_{i2} + \cdots + p_{ik}$$

$$r_n = \max(p_n, r_1 + r_{n-1}, r_2 + r_{n-2}, \dots, r_{n-1} + r_1)\,, n \geq 1$$

Para nuestro ejemplo:

Caso trivial:

$$r_1 = 1$$

Casos recursivos:

$$r_2 = \max(p_2, r_1 + r_1) = \max(5{,}2) = 5$$

$$r_3 = \max(p_3, r_1 + r_2, r_2 + r_1) = \max(8{,}6{,}6) = 8$$

$$r_4 = \max(p_4, r_1 + r_3, r_2 + r_2, r_3 + r_1) = \max(9{,}9{,}10{,}9) = 10$$

$$r_5 = \max(p_5, r_1 + r_4, r_2 + r_3, r_3 + r_2, r_4 + r_1) = \max(10{,}11{,}13{,}11) = 13$$

En general:

$$r_n = \max(p_i + r_{n-i})\,, 1 \leq i \leq n$$

La implementación en fuerza bruta de este algoritmo es de orden O(2^n).

Implementación en pseudocódigo:

```
MEMOIZED-CUT-ROD(p, n)
    Let r[0,..,n]
    for i=0 to n
        r[i]=-1
    return MEMOIZED-CUT-ROD-AUX(p,n,r)

MEMOIZED-CUT-ROD-AUX(p,n,r)
    If r[n] ≥ 0
        return r[n]
    if n == 0
        q = 0
    else q = -1
        for i = 1 to n
            q = max(q, p[i]+ MEMOIZED-CUT-ROD-AUX(p, n-1, r))
    r[n] = q
    return q
```

Ejercicio: Realizar la implementación de este ejercicio en BOTTOM-UP en C++.