1.

2.

3.

4.

5. *SumaDinámica*

5.a.

$$\text{ss}'_C s(i,j) = \begin{cases} j = 0 & \text{si } i = 0 \\ \text{ss}'_C(i-1,j) & \text{si } i \neq 0 \wedge C[i] > j \\ \text{ss}'_C(i-1,j) \vee \text{ss}'_C(i-1,j-C[i]) & \text{c.c.} \end{cases}$$

Es igual al 1.f, nada más que acotar su señoría

$O(2^n)$ llamadas recursivas son suficientes ya que son todas las combinaciones :v

5.b.

En papel

5.c.

Convencido! Es $O((n+1)(k+1)) \in O(nk)$

5.d.

$k \gg 2^n \implies O(nk) \gg O(2^n)$

$k \ll 2^n \implies O(nk) \ll O(2^n)$

5.e.

Si

6. *OptiPago*

6.a.

$$\text{cc }(B,c) = \begin{cases} (0,0) & \text{if } c = 0 \\ (\infty,\infty) & \text{if } B = \{\} \wedge c > 0 \\ (b_n, 1) & \text{if } b_n = c \\ \min_{c,q}(\text{cc}(B - \{b_n\}, c - b_n) + (b_n, 1), \text{cc}(B - \{b_n\}, c)) & \text{c.c.} \end{cases}$$

Para $n = |B|$, solve con cc $(B, c, 0)$

6.b.

Queda de ejercicio al lector

6.c.

$$\mathrm{cc}_B(i,j) = \begin{cases} (0,0) & \text{if } j = 0 \\ (\infty,\infty) & \text{if } i = 0 \land j > 0 \\ (B[i],1) & \text{if } B[i] = j \\ \min_{i,j}(\mathrm{cc}_B(i-1, j-B[i]) + (B[i],1), \mathrm{cc}_B(i-1,j)) & \text{c.c.} \end{cases}$$

solve con $\mathrm{cc}_B(|B|, c)$

6.d.

Una matriz M de $n \times c$, se crea en $O(nc)$

6.e.

Lo mismo que el **e.** pero antes de recursionar chequea si existe $M[i-1,j]$ y $M[i-1,j-B[i]]$, si no existe, lo calcula y guarda, luego lo retorna en ambos casos.

6.f.

$\mathrm{cc}_B(|B|, c)$, la complejidad es $\Omega(nc)$ y $O(nc)$, que es peor que $\Omega(1)$ pero mejor que $O(2^n)$

6.g.

Para B, c y $n = |B|$:

```
CC(i, j):
M[n × c] ← ∀i, j :: M[i][j] = (∞, ∞)
let res ← (∞, ∞)
if j = 0:
    return (0, 0)
for i ← 1 to n:
    if B[i − 1] = c:
        return (B[i − 1], 1)
    for j ← 1 to c:
        M[i][j] = min(M[i − 1][j − B[i]] + (B[i], 1), M[i − 1][j])
return M[n][c]
```

7. AstroTrade