Proyecto 1 – Diseño y Análisis de Algoritmos

- Santiago Rodriguez Mora – 202110332
- Valeria Torres Gomez – 202110363

Algoritmo de solución:

Se deben construir exactamente ***k*** números enteros no negativos cuya suma total sea ***n***, maximizando la **creatividad** total. La creatividad de una celda depende únicamente de los dígitos **3, 6, 9** presentes en cada columna decimal y de su **peso** por posición $(P_0, \ \ldots, \ P_4)$:

- Unidades P_0, decenas P_1, centenas P_2, miles P_3, diez-miles P_4
- Aportes por dígito en la columna $3 \rightarrow P_p, \ 6 \rightarrow 2P_p, \ 9 \rightarrow 3P_p$.
- El resto de los dígitos no suman creatividad.

Entradas y salidas:

E/S	Nombre	Tipo	Descripción
E	k	int	Número de celdas (números) en las que se reparte la energía. Límite: $1 \leq k \leq 10^4$.
E	n	int	Suma total de energía a repartir entre las ***k*** celdas. Límite: $1 \leq n \leq 10^5$.
E	P_0	int	Peso (creatividad) de la posición de **unidades**. Límite: $1 \leq P_0 \leq 10^5$.
E	P_1	int	Peso de la posición de **decenas**. Límite: $1 \leq P_1 \leq 10^5$.
E	P_2	int	Peso de la posición de **centenas**. Límite: $1 \leq P_2 \leq 10^5$.
E	P_3	int	Peso de la posición de **miles**. Límite: $1 \leq P_3 \leq 10^5$.
E	P_4	int	Peso de la posición de **diez-miles**. Límite: $1 \leq P_4 \leq 10^5$.
S	max_crea	int	Máximo puntaje de creatividad posible.

Algoritmo DP:

Podemos definir como $n = \sum_{p \geq 0} n_p \, 10^p$ su descomposición decimal y *k* el número de celdas. En la columna p, la suma de dígitos entre las *k* celdas es:

$$S_p = \sum_{j=1}^{k} d_{p,j} \in [0,9k]$$

Esto cumple la conservación de suma con acarreos: $S_p + c_p = n_p + 10c_{p+1} \Leftrightarrow S_p = n_p - c_p + 10c_{p+1}$. Además, la creatividad **local** de una columna depende de cuántas unidades de 3 caben en S_p:

$$\text{crea}_p(S_p) = P_p \left\lfloor \frac{S_p}{3} \right\rfloor$$

Procesamos columnas desde la más significativa hasta la de unidades (cubriendo al menos hasta $p = 4$). En cada columna mantenemos un arreglo

Proyecto 1 – Diseño y Análisis de Algoritmos

$$\text{dp[c]} = \textbf{max\_crea}\ \textit{si entran c acarreos a la columna actual}$$

- **Base:** más allá de la última columna no debe quedar carry: $\text{dp}_{\text{p+1}} = [0]$.

Transiciones:

Para la columna p con un dígito $t = n_p$ y peso $w = P_p$ (si $p > 4$, $w = 0$):

1. Definir rangos de candidatos para $y = c_{p+1}$(carry saliente), dados c (carry entrante):

$$l = \left\lceil \frac{c - t}{10} \right\rceil, \qquad u = \left\lfloor \frac{9k + c - t}{10} \right\rfloor, \qquad y \in \{l, u\} \cap [0, \text{limiteSig}\,]$$

2. Descomponer en mod 3: $c = 3q + r$ con $r \in \{0,1,2\}$. Usando

$$\left\lfloor \frac{n_p - c + 10y}{3} \right\rfloor = 3y + \left\lfloor \frac{y + t - r}{3} \right\rfloor - \left\lfloor \frac{c}{3} \right\rfloor$$

   la contribución local se separa en una parte que no depende del c específico (dentro de la clase r) y una corrección $-\left\lfloor \frac{c}{3} \right\rfloor$.

3. Pre-cómputo por residuo: Para cada $r \in \{0,1,2\}$ construimos:

$$A_r[y] = \text{dp}_{\text{sig}}[y] + 3wy + w\left\lfloor \frac{y + t - r}{3} \right\rfloor$$

4. Sliding window (deque):
   a. Para cada clase r, recorremos los c tales que $c = r (\text{mod } 3)$.
   b. Para cada c calculamos $[\ell, u]$ y obtenemos en $O(1)$ suprimiendo el máximo de $A_r[y]$ en esa ventana usando un deque monótono (cada índice entra y sale a lo sumo una vez).

5. Actualización del estado:

$$\text{dp}_{\text{sig}}[c] = \max A_r[y] - w\left\lfloor \frac{c}{3} \right\rfloor$$

   Si no hay un y válido, definimos un PUNTAJE_MINIMO.

6. Pasar a la siguiente columna. $\text{dp}_{\text{sig}} \leftarrow \text{dp}$.
7. Respuesta final: Tras procesar todas las columnas, la solución del caso es $\max(0, \text{dp}_{\text{sig}}[0])$.

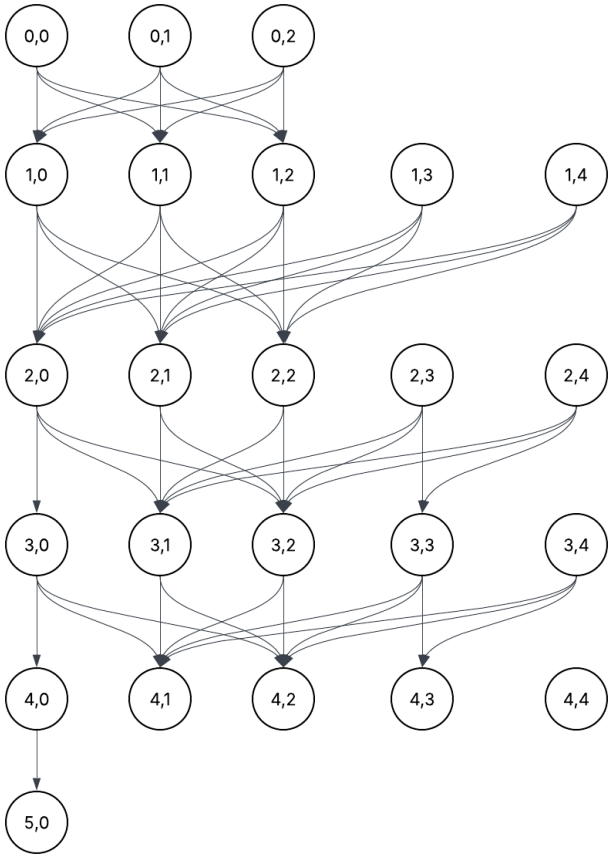
Ecuación de recurrencia:

$$\text{DP[p][c]} = \max_{y \in \{l,u\}} \left\{ w_p \left\lfloor \frac{n_p - c + 10y}{3} \right\rfloor + \text{DP}[p+1][y] \right\}$$

La respuesta final sería DP[0][0]

Proyecto 1 – Diseño y Análisis de Algoritmos

Grafo de necesidades:



Análisis de complejidades espacial y temporal:

Sea $L = \max(4, [\log_{10} n])$ el número de columnas consideradas y $C = \mathcal{O}(k)$ el rango operativo de acarreos.

a) Tiempo:

Por columna:

- Pre-cómputo $A_r[y]$ para $r = 0,1,2 \rightarrow \mathcal{O}(C)$
- Para cada r , recorrido de $c = r \ (mod\ 3)$ con deque (cada y entra/sale máximo una vez): $\mathcal{O}(C)$

En L columnas:

- $T(n,k) = \mathcal{O}(L \cdot C) = \mathcal{O}(\log n \cdot k)$

b) Espacio:

Dos capas de DP y tres arreglos A de orden C :

$$MEM(n,k) = \mathcal{O}(C) = \mathcal{O}(k)$$