Ayudantía 9 - Algoritmos y Complejidad
Generativas - Diseño - Montecarlo

Sassy complexers

1. Generativas - *La revancha*

Recuerde que para resolver recurrencias tiene varios metodos:

1. Expandir la expresion hasta que se le ocurra algo
2. Dibujar el árbol de recursion hasta que se le ocurra algo
3. Teorema maestro (solo si la recurrencia es de la forma $aT(n/b) + f(n)$)
4. Metodo de Akra-bazzi (Prefiera generatrices antes que esto)
5. Funciones Generatrices

Así que no se acompleje si no logra captar alguno de estos por que alternativas existen, hoy trataremos de revisar uno de los mas utiles y temidos.

Este es el reciclado máximo de viejas ayudantias de estructuras discretas, las cuales niguno de sus ayudantes entiende bien, pero vamos a intentar que se entienda aunque no entiendan.

Por si no se ha dado cuenta, un polinomio se puede almacenar como un arreglo de coeficientes, donde el indice del coeficiente marca el grado al que está elevado el x al que acompaña, es decir:

$$a_0 + a_1 x + a_2 x^2 + \cdots + a_n x^n \leftrightarrow (a_0, a_1, a_2, \cdots, a_n)$$

Esto también se puede realizar al revés, si se tiene una serie de n valores se pueden "almacenar" en un polinomio de grado $n+1$ y trabajaremos con estos polinomios por las siguientes razones:

1. **La estructura que ofrecen los polinomios** no es tan rígida como la de los vectores y, a diferencia de los conjuntos, aseguran un orden. Podemos asignar cualquier valor en cualquier posición $i \geq 0, i \in Z_0^+$, por lo que disponemos de infinitas posiciones.
   Más aun, si la serie tiene infinitos números, y no converge para algunos valores de z la expresión sigue siendo válida, pues utilizamos las potencias de z sólo para separar los términos.
   Notacionalmente, si nuestra secuencia es $\langle a_n \rangle_{n\geq 0}$ esta idea se escribe de la siguiente manera:

$$\langle a_n \rangle_{n\geq 0} \stackrel{\text{ogf}}{\longleftrightarrow} \sum_{n\geq 0} a_n z^n$$

2. **Pueden compactarse**, si hay muchos o infinitos términos; utilizando fórmulas ya conocidas, podemos "compactarlos" en una fórmula mucho más fácil de manejar.
   Por ejemplo, si tenemos $a_n = 2n+1$:

$$\sum_{n\geq 0} a_n z^n = \sum_{n\geq 0} (2n+1) z^n$$

   Y sabemos que, si $|z| < 1$:

$$\sum_{n\geq 0} z^n = \frac{1}{1-z} \qquad \text{(Serie geometrica)}$$

   ... y al derivar la expresión anterior:

$$\begin{aligned} \sum_{n\geq 0} n z^{n-1} &= \frac{1}{(1-z)^2} && \text{(Primera derivada de la serie geometrica)} \\ \sum_{n\geq 0} (n+1) z^n &= \frac{1}{(1-z)^2} && \text{(Corrimos el } n \text{ uno hacia adelante)} \end{aligned}$$

   Podemos "compactar" nuestra sumatoria original:[1]

$$\begin{aligned} \sum_{n\geq 0} (2n+1) z^n &= \sum_{n\geq 0} (2(n+1)-1) z^n && \text{(Suma un cero conveniente)} \\ &= 2\sum_{n\geq 0} (n+1) z^n - \sum_{n\geq 0} z^n && \text{(Queda una resta de geometricas)} \\ &= \frac{1}{(1-z)^2} - \frac{1}{1-z} \\ &= \frac{z}{(1-z)^2} \end{aligned}$$

   El "arte" de funciones generatrices está en pasar los polinomios a esta forma compacta y vice-versa, esto último para recobrar los coeficientes luego de hacer operaciones sobre el polinomio (por ejemplo, al elevarlo al cuadrado), que serían imposibles de realizar en su forma de sumatoria.

3. **Sus propiedades combinatoras**. Supongamos que quiere ver cuantas maneras hay de juntar 3 manzanas, si tiene 3 manzanas verdes y 2 manzanas rojas.

   Al observar las manzanas verdes, ve que tiene:

   - 1 forma de elegir ninguna manzana verde. (nada)
   - 1 forma de elegir 1 manzana verde. (Verde)
   - 1 forma de elegir 2 manzanas verdes. (Verde, Verde)
   - 1 forma de elegir 3 manzanas verde. (Verde, Verde, Verde)

   Esto se puede hacer equivaler al polinomio:

$$V(z) = 1\cdot z^0 + 1\cdot z^1 + 1\cdot z^2 + 1\cdot z^3$$

   Donde utilizamos la posición en la secuencia de números, vale decir, el exponente del z o el índice en el arreglo para indicar el *tamaño* o cantidad de algo que aporta una elección.

[1] Este resultado debe revisarse, puede estar malo

De la misma manera, para las manzanas rojas tendríamos:

$$R(z) = 1 \cdot z^0 + 1 \cdot z^1 + 1 \cdot z^2$$

Y para resolver lo que se nos pidió en un principio tenemos que ver las cantidad de formas posibles de lograr el exponente z^3 con las opciones que se tienen para elegir manzanas verdes y rojas, para esto, no tenemos que hacer nada más que multiplicar ambos polinomios.

$$\begin{aligned} V(z) \cdot R(z) &= (1 + z + z^2 + z^3)(1 + z + z^2) \\ &= 1 + 2z + 3z^2 + 3z^3 + 2z^4 + 1z^5 \end{aligned}$$

Vemos que el coeficiente que acompaña al z^3 es 3, por lo tanto hay 3 formas de juntar 3 manzanas con las manzanas rojas y verdes que nos dieron, y no sólo eso, también descubrimos que:

- Hay 1 forma de elegir 0 manzanas.
- Hay 2 forma de elegir 1 manzana.
- Hay 3 forma de elegir 2 manzanas.
- Hay 2 forma de elegir 4 manzanas.
- Sólo hay 1 forma de elegir 5 manzanas.

Esto, que podría parecer simple, se convierte en una herramienta muy poderosa cuando tratamos con polinomios infinitos y los "compactamos", multiplicamos y luego "descompactamos" la expresión resultante, o términos más coloquiales, "la hacemos confesar".

4. **Nos sirven para revelar secuencias a partir de recurrencias** (y esto es a lo que venimos a Algoco™), por ejemplo, desconocemos una formula directa para los términos a_n de una secuencia, pero nos dan la fórmula recurrente:

$$\begin{aligned} a_0 &= 1 \qquad \text{(Caso Base)} \\ a_{n+1} &= 2a_n + 1 \qquad \text{(Caso Recurrente)} \end{aligned}$$

Nuestra única manera de obtener, de manera formal, una fórmula directa para a_n es considerar infinitas veces la segunda expresión (para cada valor de n) esto lo podemos hacer si tomamos esta expresión y la multiplicamos por z^n y sumamos sobre todos los valores de n, para que resulten varias generatrices.

$$\begin{aligned} a_{n+1} &= 2a_n + 1 \\ a_{n+1}z^n &= 2a_n z^n + z^n \qquad \text{(Infinitos } a_n\text{'s que dependen de } a_n\text{'s anteriores)} \\ \sum_{n\geq 0} a_{n+1}z^n &= \sum_{n\geq 0} 2a_n z^n + \sum_{n\geq 0} z^n \end{aligned}$$

Ahora, en esta expresión, intentamos despejar la generatriz asociada a nuestra secuencia, es decir:

$$A(z) = \sum_{n\geq 0} a_n z^n \qquad \text{(Por que queremos el } a_n\text{, cierto?)}$$

Haga el algebra:

$$\sum_{n\geq 0} a_{n+1} z^n = \sum_{n\geq 0} 2a_n z^n + \sum_{n\geq 0} z^n$$

Sacamos la constante de la suma del medio, entonces aparece el $A(z)$ que buscamos

$$\sum_{n\geq 0} a_{n+1} z^n = 2A(z) + \sum_{n\geq 0} z^n$$

Corremos el indice en la suma de la izquierda, nos queda un z^{-1} entremedio

$$\begin{aligned}
\sum_{n\geq 1} a_n z^{n-1} &= 2A(z) + \sum_{n\geq 0} z^n \\
\sum_{n\geq 1} a_n z^n z^{-1} &= 2A(z) + \sum_{n\geq 0} z^n \\
\frac{\sum_{n\geq 1} a_n z^n}{z} &= 2A(z) + \sum_{n\geq 0} z^n
\end{aligned}$$

Sacamos el caso base para volver a correr el índice, aparecio un salvaje $A(z)$

$$\begin{aligned}
\frac{\sum_{n\geq 0} a_n z^n - a_0}{z} &= 2A(z) + \sum_{n\geq 0} z^n \\
\frac{A(z) - a_0}{z} &= 2A(z) + \sum_{n\geq 0} z^n \\
\frac{A(z) - 1}{z} &= 2A(z) + \sum_{n\geq 0} z^n \qquad (a_0 = 1, \text{Caso base}) \\
\frac{A(z) - 1}{z} &= 2A(z) + \frac{1}{1-z} \qquad (\text{Serie geometrica conocida}) \\
A(z) - 1 &= 2zA(z) + \frac{z}{1-z} \qquad (\text{Multiplica el z del denominador de la izquierda}) \\
(1-2z)A(z) - 1 &= \frac{z}{1-z} \qquad (\text{Pasamos } 2zA(z) \text{ a la izquierda y factorizamos}) \\
(1-2z)A(z) &= 1 + \frac{z}{1-z} \qquad (\text{Ahora pasamos el } -1) \\
(1-2z)A(z) &= \frac{1-z}{1-z} + \frac{z}{1-z} \qquad (\text{Suma unos convenientes}\ldots) \\
(1-2z)A(z) &= \frac{1}{1-z} \qquad (\ldots\text{Magia negra del tipo oscuro}) \\
A(z) &= \frac{1}{(1-z)(1-2z)}
\end{aligned}$$

Ahora sólo nos queda extraer los coeficientes del $A(z)$, es decir, *hacerlo confesar*. Puesto que no tenemos ninguna fórmula de sumatorias que sea semejante, va a haber que descomponerlo en fórmulas que si conozcamos.

Para hacerlo utilizamos **Fracciones Parciales**.

$$\begin{aligned}\frac{1}{(1-z)(1-2z)} &= \frac{-1}{1-z} + \frac{2}{1-2z} \\ &= -\sum_{n\geq 0} z^n + 2\sum_{n\geq 0} 2^n z^n \\ &= -\sum_{n\geq 0} z^n + \sum_{n\geq 0} 2^{n+1} z^n \\ &= \sum_{n\geq 0} (2^{n+1}-1) z^n\end{aligned}$$

Finalmente, llegamos a la conclusión de que los a_n son:

$$a_n = 2^{n+1} - 1$$

Y así se resuelven recurrencias usando generatrices.

2. Montecarlo - La introducción mas fome que podrá encontrar

Despues de pasar tantos años en esta universidad se puede haber dado cuenta de que tirando las respuestas al *achunte* tiene mas posibilidades de lograr un buen resultado que estudiando seriamente. Por ejemplo, calculemos:

$$\int_0^1 \int_0^1 \sqrt{x^2+y^2}\, dy dx$$

Sabemos que es $\frac{\Pi}{4}$ cierto? es la integral del cuarto de circulo. No pierda el tiempo con algebra, tirela al achunte.

1. Dibuje un cuadrado al rededor de este cuarto de circulo.
2. Lanze N dardos a este cuadrado
3. Cuente cuantos cayeron dentro del cuarto de circulo
4. Su integral es la razón entre los dardos que cayeron dentro con el total multiplicado por el Area del cuadrado $\frac{n_{dentro}}{N} \cdot A$

Listing 1: Montecarlo simulating PI

```
import numpy as np
import numpy.random as r

N_points = 1000000

count = 0
for x, y in r.uniform(0,1, (N_points,2)):
    if x*x+y*y < 1:
        count +=1

print("Pi/4:", count/N_points)
print("Pi:", 4*(count/N_points))
```

3. Diseño (de algoritmos)

De aquí en adelante son ejercicios no mas

1. La operación *shuffle* consiste en generar una permutación aleatoria de los elementos de un array *a* de manera que todas las combinaciones sean equiprobables. A continuación se entregan algunos algoritmos para lograr esto [2], indique cómo funcionan y la complejidad de cada uno:

   a) Este algoritmo trabaja con pares [3] :

   Listing 2: Primero

```c
typedef struct{
  int r;
  int v;
} pair;

pair pairs[n];
for(int i=0;i<n;i++){
  pairs[i].r = rand();
  pairs[i].v = a[i];
}
qsort(pairs, n, sizeof(pair), sort_pairs_by_r)
for(int i=0;i<n;i++){
  a[i] = pairs[i].v;
}
```

   b) Este algoritmo está basado en una historia real de AlgoCo [4]:

   Listing 3: Segundo

```c
int rema[n];
int rema_n = n;
for(int i=0;i<rema_n;i++){
  rema[i] = i;
}
for(int i=0;i<n;i++){
  int nind = rand()%rema_n;
  int next = rema[nind];
  af[i] = a[next];
  //
  for(k=nind;k<rema_n-1;k++){
    rema[k] = rema[k+1];
  }
  rema_n--;
}
```

[2] Suponiendo que la función `rand` entrega un entero aleatorio positivo en un rango muy grande.
[3] Asuma que la función `sort_pairs_by_r` ordena elementos de acuerdo al valor `r` del par.
[4] Asuma que `af` será el nuevo arreglo `a`

c) Este algoritmo es lo que se le ocurrió al ayudante después de ver lo anterior:

Listing 4: Tercero

```
int rema[n];
int rema_n = n;
for(int i=0;i<n;i++){
  rema[i] = i;
}
for(int i=0;i<n;i++){
   int nind = rand()%rema_n;
   int next = rema[nind];
   af[i] = a[next];
   //
  rema[nind] = rema[rema_n-1]
  rema_n--;
}
```

d) Este último es conocido como el algoritmo de Fisher-Yates o Knuth shuffle [5]:

Listing 5: Cuarto

```
for(int i=n-1;i>=1;i--){
  j = rand()%(i+1)
  aux = a[j]
  a[j] = a[i]
  a[i] = aux
```

e) Este otro es interesante de analizar:

Listing 6: Quinto

```
int used[n];
for(int i=0;i<n;i++){
  used[i] = 0;
}
used_n = 0;
while(used_n<n){
  int next = rand()%n;
  if(!used[next]){
    af[used_n] = a[next];
    used[next] = 1;
    used_n++;
  }
}
```

La complejidad del primer algoritmo está dada por $O(n \log n)$ puesto que hay una función de ordenamiento, el segundo algoritmo tiene complejidad $O(n^2)$, porque en peor caso requiere hacer $n(n+1)/2$ movimientos para ajustar `rema`. El tercer algoritmo es una mejora

[5] https://en.wikipedia.org/wiki/Fisher%E2%80%93Yates_shuffle

potente del segundo, tiene complejidad $O(n)$, al igual que el cuarto, que además tiene la ventaja de no tener que crear un array en paralelo.

El quinto algoritmo, ante mala suerte persistente puede no terminar nunca, por lo que no podemos encontrar una cota superior para el peor caso, (notemos que el algoritmo va a tener una probabilidad de $1/n$ de encontrar en cada iteración el último elemento de `a` no utilizado). Podemos realizar un análisis de probabilidades para analizar cuanto demoraría, y podría usar sus conocimientos de Estadística Computacional [6]. No obstante, si se quisiera, por ejemplo, obtener una permutación aleatoria de una porción de elementos de `a` (e.g. la mitad), en promedio puede terminar en un tiempo más que aceptable y su uso puede estar justificado.

[6] Sería una buena forma de hacer el paso de *shuffle* para complementar *bogosort*.

Torpedo Oficial

- Fórmulas $\frac{1}{1-z}$ y derivadas:

$$\begin{aligned}
\frac{1}{1-z} &= \sum_{n\geq 0} z^n \\
\frac{1}{(1-z)^2} &= \sum_{n\geq 0} (n+1) z^n \\
\frac{1}{(1-z)^3} &= \sum_{n\geq 0} \frac{(n+2)(n+1)}{2} z^n \\
\vdots \quad &= \quad \vdots \\
\frac{1}{(1-z)^k} &= \sum_{n\geq 0} \frac{(n+1)^{\overline{k-1}}}{(k-1)!} z^n
\end{aligned}$$

- Teorema del binomio:

$$(1+z)^\alpha = \sum_{n\geq 0} \binom{\alpha}{n} z^n$$

Se suele utilizar cuando $\alpha < 0$, es decir, cosas de la forma:

$$\frac{1}{(1+z)^\beta} = (1+z)^{-\beta}$$

Entonces:

$$\frac{1}{(1+z)^\beta} = \sum_{n\geq 0} \binom{-\beta}{n} z^n$$

sin olvidar que:

$$\binom{-\beta}{n} = (-1)^n \binom{\beta+n-1}{\beta-1}$$

También suele ocurrir que z esté restando en vez de sumando. En ese caso:

$$(1-z)^\gamma = (1+(-z))^\gamma = \sum_{n\geq 0} \binom{\gamma}{n} (-z)^n = \sum_{n\geq 0} \binom{\gamma}{n} (-1)^n z^n$$

- Es importante notar que se puede reemplazar $z = cz$ en las fórmulas anteriores y obtener fórmulas nuevas, el ejemplo más importante de esto es:

$$\frac{1}{1-cz} = \sum_{n\geq 0} c^n z^n$$

- Extraer el primer término reemplazando $z = 0$.

$$A(0) = a_0$$

- Obtener la suma de todos los términos reemplazando $z = 1$:

$$A(1) = \sum_{n\geq 0} a_n$$

Puede resultar ser infinito.

- Extraer un término a_i específico derivando i veces y reemplazando $z = 0$.

$$a_i = \frac{A^{(i)}(0)}{i!}$$

- Notación $[z^n]$ (indica que se busca extraer el término que acompaña z^n en la expresión).

$$a_n = [z^n]\, A(z)$$

  Tiene la siguientes propiedades:

$$[z^n]\, z^k A(z) = [z^{n-k}]\, A(z)$$

$$[z^n](A(z) + B(z)) = [z^n]A(z) + [z^n]B(z)$$

- Corrimiento de índices:

$$\begin{aligned}\sum_{n\geq 0} a_{n+k} z^n = \frac{\sum_{n\geq 0} a_{n+k}\, z^{n+k}}{z^k} &= \frac{\sum_{n\geq k} a_n\, z^n}{z^k} = \frac{\sum_{n\geq 0} a_n\, z^n - \sum_{n<k} a_n\, z^n}{z^k} \\ &= \frac{A(z) - \sum_{n<k} a_n z^n}{z^k}\end{aligned}$$

Easter Egg: Copy-Paste de una ayudantia de discretas

1. Para llenar una canasta se debe colocar:

   - Una cantidad par de naranjas.
   - Cualquier cantidad de manzanas, donde estas pueden ser rojas o verdes.

   ¿Cuántas maneras hay de llenar una canasta en que caben exactamente 1999 frutas?

   Desarrollo Se plantean las generatrices de cada grupo de frutas, la de las naranjas es sencilla pues sólo hay 1 posibilidad de formar cada cantidad par de naranjas y 0 de formar cada impar:

$$\begin{aligned}N(z) &= 1z^0 + 0z^1 + 1z^2 + 0z^3 + 1z^4 + 0z^5 + 1z^6 + \cdots \\ &= 1 + z^2 + z^4 + z^6 + \cdots \\ &= \sum_{n\geq 0} 1 \cdot z^{2n} \\ &= \frac{1}{1-z^2} \\ &= \frac{1}{(1+z)(1-z)}\end{aligned}$$

Para una cantidad n de manzanas hay $\left(\!\binom{2}{n}\!\right) = \binom{n+2-1}{n} = \frac{(n+1)!}{n!\cdot 1!} = n+1$ formas de elegirlas. La generatriz queda:

$$
\begin{aligned}
M(z) &= 1z^0 + 2z^1 + 3z^2 + 4z^3 + 5z^4 + \cdots \\
&= 1 + 2z + 3z^2 + 4z^3 + 5z^4 + \cdots \\
&= \sum_{n\geq 0} (n+1)z^n \\
&= \frac{1}{(1-z)^2}
\end{aligned}
$$

Lo anterior se deduce derivando:

$$
\begin{aligned}
\frac{1}{1-z} &= \sum_{n\geq 0} z^n \\
\frac{1}{(1-z)^2} &= \sum_{n\geq 1} nz^{n-1} \\
&= \sum_{n\geq 0} (n+1)z^n
\end{aligned}
$$

Ahora se multiplican ambas generatrices para obtener la generatriz del total de frutas:

$$
\begin{aligned}
M(z)\cdot N(z) &= \frac{1}{(1-z)^2}\cdot\frac{1}{(1-z)(1+z)} \\
&= \frac{1}{(1-z)^3(1+z)}
\end{aligned}
$$

Se utilizan fracciones parciales...

$$
\frac{1}{8(1+z)} + \frac{1}{8(1-z)} + \frac{1}{4(1-z)^2} + \frac{1}{2(1-z)^3}
$$

Podemos leer los coeficientes de z^n en esto:

$$
\begin{aligned}
\frac{1}{8}(-1)^n + \frac{1}{8} + \frac{1}{4}(-1)^n\binom{-2}{n} + \frac{1}{2}(-1)^n\binom{-3}{n} \\
&= \frac{(-1)^n}{8} + \frac{1}{8} + \frac{1}{4}\binom{n+2-1}{2-1} + \binom{n+3-1}{3-1} \\
&= \frac{(-1)^n}{8} + \frac{1}{8} + \frac{1}{4}\,\frac{n+1}{1!} + \frac{(n+2)(n+1)}{2!} \\
&= \frac{4n^2 + 14n + 11 + (-1)^n}{8}
\end{aligned}
$$

Finalmente, el coeficiente de z^{1999} es 2 001 500.

2. Se tiene la siguiente recursión:

$$a_n = 2a_{n-1} + 3n \quad (n \geq 1); \qquad a_0 = 2$$

Extraer una fórmula directa para cada a_n.

Desarrollo Para trabajar más fácilmente se corren los índices:

$$a_{n+1} = 2a_n + 3(n+1) \quad (n \geq 0)$$

Se define la generatriz a buscar:

$$A(z) = \sum_{n\geq 0} a_n z^n$$

Aplicando las propiedades, con el valor inicial y despejando:

$$\begin{aligned}
\frac{A(z) - a_0}{z} &= 2A(z) + 3\frac{1}{(1-z)^2} \\
\frac{A(z) - 2}{z} &= 2A(z) + 3\frac{1}{(1-z)^2} \\
A(z) &= \frac{2z^2 - z + 2}{(1-z)^2(1-2z)}
\end{aligned}$$

Si hay tiempo desarrollar en fracciones parciales:

$$A(z) = \frac{8}{1-2z} - \frac{3}{1-z} - \frac{3}{(1-z)^2}$$

Se extraen los coeficientes:

$$\begin{aligned}
[z^n]A(z) &= 8 \cdot 2^n - 3 - 3\binom{-2}{n} \\
&= 8 \cdot 2^n - 3 - 3\binom{n+2-1}{2-1} \\
&= 8 \cdot 2^n - 3 - 3\frac{n+1}{1!} \\
&= 8 \cdot 2^n - 3 - 3\frac{n+1}{1!} \\
&= 2^{n+3} - 3n - 6
\end{aligned}$$

3. Para la realización de un software se dispone de tres ingenieros, cinco programadores y seis practicantes, cada uno de estos tiene el siguiente rendimiento:

Cargo	Cantidad disponible	Rendimiento
Ingeniero	3	4
Programador	5	3
Practicante	6	2

   a) Si todos los trabajadores del mismo cargo son iguales ¿Cuantos posibles grupos de trabajo se pueden lograr con un rendimiento de 13 unidades de rendimiento? Exprese la solución utilizando funciones generatrices.

   Se utilizan los exponentes de z para marcar la cantidad de unidades de rendimiento (pues eso es lo que se busca sumar).

   Ahora se plantean las generatrices de cada una de los cargos:

   Para los ingenieros, puesto que son iguales sólo tienen una forma (porque sólo importa el número) de lograr 0, 4, 8 y 12 unidades de rendimiento.

$$I(z) = 1 + z^4 + z^8 + z^{12}$$

   Para los programadores, puesto que son iguales, sólo tienen una forma de lograr 0, 3, 6, 9, 12 y 15 unidades de rendimiento.

$$P(z) = 1 + z^3 + z^6 + z^9 + z^{12} + z^{15}$$

   Finalmente para los practicantes:

$$X(z) = 1 + z^2 + z^4 + z^6 + z^8 + z^{10} + z^{12}$$

   Hay que extraer el coeficiente que acompaña z^{13} de la generatriz:

$$I(z)P(z)X(z)$$

   Lo que se expresa como:

$$[z^{13}]I(z)P(z)X(z)$$

   Si alcanza el tiempo, desarrollar los polinomios en su forma compacta.

$$\begin{aligned} z^{39} &+ z^{37} + z^{36} + 2z^{35} + z^{34} + 3z^{33} + 2z^{32} + 4z^{31} + 3z^{30} + 5z^{29} + 4z^{28} + 7z^{27} \\ &+ 5z^{26} + 7z^{25} + 7z^{24} + 8z^{23} + 7z^{22} + 8z^{21} + 8z^{20} + 8z^{19} + 8z^{18} + 7z^{17} \\ &+ 8z^{16} + 7z^{15} + 7z^{14} + 5z^{13} + 7z^{12} + 4z^{11} + 5z^{10} + 3z^9 + 4z^8 \\ &+ 2z^7 + 3z^6 + z^5 + 2z^4 + z^3 + z^2 + 1 \end{aligned}$$

   El coeficiente resulta ser 5.

b) ¿Qué modificación haría al desarrollo si se considera que los programadores son diferentes?

Los coeficientes de $P(z)$ ya no serán 1 solamente, pues habrán diferentes formas de obtener un número fijo de programadores.

$$P(z) = \binom{5}{0}1 + \binom{5}{1}z^3 + \binom{5}{2}z^6 + \binom{5}{3}z^9 + \binom{5}{4}z^{12} + \binom{5}{5}z^{15}$$

c) ¿Qué modificación haría al desarrollo si se exige que en el proyecto participen al menos dos ingenieros y una cantidad par de practicantes?

En la generatriz de los ingenieros pasarían a ser 0 los coeficientes que involucran 0 o 1 ingeniero.

$$I(z) = z^8 + z^{12}$$

En la generatriz de los practicantes, los coeficientes que involucran una cantidad impar de practicantes pasan a ser 0.

$$X(z) = 1 + z^4 + z^8 + z^{12}$$

4. Considere los valores de la siguiente expresión:

$$\alpha_1 + 2\alpha_2 + 3\alpha_3$$

donde $a_i \in \{0, 1, 2\}$

Utilice generatrices para determinar cuantas maneras hay de sumar 6.

Pro-Tip: Recuerde que aquí lo que se quiere sumar son unidades en la ecuación. Hay que plantear 3 generatrices, de acuerdo con lo que aporta cada a_i.

Solución

$$A_1(z) = 1 + z + z^2$$
$$A_2(z) = 1 + z^2 + z^4$$
$$A_3(z) = 1 + z^3 + z^6$$

Luego, la generatriz de la suma resulta:

$$\begin{aligned} A_1(z)A_2(z)A_3(z) &= (1 + z + z^2)(1 + z^2 + z^4)(1 + z^3 + z^6) \\ &= z^{12} + z^{11} + 2z^{10} + 2z^9 + 3z^8 + 3z^7 + 3z^6 + 3z^5 \\ &\quad + 3z^4 + 2z^3 + 2z^2 + z + 1 \end{aligned}$$

Como el coeficiente que acompaña z^6 es 6, hay 6 maneras diferentes de sumar 6 con la fórmula y los posibles valores para cada a_i.