Daniel Alejandro Osornio López
0244685@up.edu.mx

29 / Sep / 2023

Actividad 3.4

1. Desarrollar los binomios de cada uno de los siguientes incisos, usando para ello el teorema binomial.
   - $(4x^3 - 2y)^3$

     $= \binom{3}{0}(4x^3)^{3-0}(-2y)^0 + \binom{3}{1}(4x^3)^{3-1}(-2y)^1 + \binom{3}{2}(4x^3)^{3-2}(-2y)^2 + \binom{3}{3}(4x^3)^{3-3}(-2y)^3$

     $= 1(4x^3)^3(-2y)^0 + 3(4x^3)^2(-2y)^1 + 3(4x^3)^1(-2y)^2 + 1(4x^3)^0(-2y)^3$

     $= 64x^9 - 96x^6y + 48x^3y^2 - 8y^3$
   - $(x^2 + 3y^2)^4$

     $= \binom{4}{0}(x^2)^{4-0}(3y^2)^0 + \binom{4}{1}(x^2)^{4-1}(3y^2)^1 + \binom{4}{2}(x^2)^{4-2}(3y^2)^2$

     $+\binom{4}{3}(x^2)^{4-3}(3y^2)^3 + \binom{4}{4}(x^2)^{4-4}(3y^2)^4$

     $= 1(x^2)^4(3y^2)^0 + 4(x^2)^3(3y^2)^1 + 6(x^2)^2(3y^2)^2 + 4(x^2)^1(3y^2)^3 + 1(x^2)^0(3y^2)^4$

     $= x^8 + 12x^6y^2 + 54x^4y^4 + 108x^2y^6 + 81y^8$

2. Determina el coeficiente de:
   - xyz^2 en la expansión $(x + y + z)^4$

     $\frac{4!}{1!1!2!} = 12$
   - $w^3x^2yz^2$ en la expansión $(2w - x + 3y - 2z)^8$

     $\frac{8!}{3!2!1!2!} = 1680$

3. Identifica el método de conteo y resuelve lo que solicita en cada situación (Combinación sin o con repetición, Permutación sin o con repetición, principio multiplicativo).
   - Cuando se arrojan simultáneamente 4 monedas, ¿Cuáles son los resultados posibles que se pueden obtener?

     2 times 2 times 2 times 2 = 16
   - Determina el número de soluciones enteras de $x_1 + x_2 + x_3 + x_4 = 32$

$$r = 32$$
$$n = 4$$
$$\binom{4 + 32 - 1}{32} = 6545$$

4. Recrea el siguiente código con base al método de Sort de burbuja, y explica a detalle lo que realiza cada línea del código

```
/// Para ordenar cualquier arreglo de tipos T necesitamos que implementen la
/// funcionalidad que permite comparar entre dos valores de T. Por eso requiere
/// la característica `Ord`
fn sort<T: Ord>(arr: &mut [T]) {
    // Para cada elemento en el arreglo
    for i in 0..arr.len() - 1 {
        // Vamos ordenando poco a poco las secciones
        // de forma que hay que revisar menos valores en la siguiente iteración
        // pues los elementos fueron ordenados el iteraciones pasadas
        for j in 0..arr.len() - i - 1 {
```

```
            if arr[j] > arr[j + 1] {
                // Si el valor es efectivamente, mayor, los intercambiamos
                // de forma que se vayan acomodando poco a poco
                arr.swap(j, j + 1);
            }
        }
    }
}
```