

Formulas

1. Números de Catalán:

$$C[n] \Rightarrow FOR(k = 0, n - 1) \ C[k] * C[n - 1 - k]$$

$$C[n] \Rightarrow Comb(2 * n, n) / (n + 1)$$

$$C[n] \Rightarrow C[n - 1] * (4 * n - 2) / (n + 1)$$

2. Número Desarranjo:

$$d(1) = 0, \ d(2) = 1; \ d(n) = (n - 1) * (d(n - 1) + d(n - 2))$$

3. Que cumpla al menos k requisitos: $\sum_{i=0}^{n-k} (-i)^i \binom{k+i-1}{k-1} S_{i+k}$

4. Que cumpla k requisitos: $\sum_{i=0}^{n-k} (-i)^i \binom{k+i}{k} S_{i+k}$

5. Distancia en movimiento rectilíneo uniforme acelerado:

$$X(t) = x_0 + v_0 * t + \frac{a * t^2}{2}$$

6. Iterar por todas las submascaras de n : $for(i = n; !i; i = n \& (i - 1))$

7. Un grafo es planar si: $n - m + f = 2$, entonces $f \leq 2n - 4$, $m \leq 3n - 6$ y contiene un vertice con grado ≤ 5 .

$$8. \sum_{i=1}^n i^2 = \frac{n(n+1)(2n+1)}{6}, \sum_{i=1}^n i^3 = \frac{n^2(n+1)^2}{4},$$

Solution Ideas

1. Dynamic Optimizations:

■ Convex hull optimization:

$$* dp[i] = \min_{j < i} \{ dp[j] + b[j] * a[i] \}$$

$$* b[j] \geq b[j + 1], \ a[i] \leq a[i + 1]$$

$$* O(n^2) \text{ to } O(n)$$

■ Divide and conquer optimization:

$$* dp[i][j] = \min_{k < j} \{ dp[i - 1][k] + C[k][j] \}$$

$$* opt[i][j] \leq opt[i][j + 1]$$

$$* O(kn^2) \text{ to } O(kn \log n)$$

$$* \text{sufficient: } C[a][c] + C[b][d] \leq C[a][d] + C[b][c], \ a \leq b \leq c \leq d \ (QI)$$

■ Divide and conquer optimization:

$$* dp[i][j] = \min_{i < k < j} \{ dp[i][k] + dp[k][j] + C[i][j] \}$$

$$* opt[i][j - 1] \leq opt[i][j] \leq opt[i + 1][j]$$

$$* O(n^3) \text{ to } O(n^2)$$

$$* \text{sufficient: } (QI) \text{ and } C[b][c] \leq C[a][d], \ a \leq b \leq c \leq d$$

2. Optimizations

- Use bitset (/64)

- Switch order of loops (cache locality)

3. Procesar queries offline (Mo's algorithm)

4. Sqrt decomposition

5. Meet in the middle

6. Fast Fourier Transform

7. Buscar cota extrema

8. Flujo

Debugging Tips

1. Stack overflow? Recursividad del Dfs.

2. Redondeando números negativos?

3. Double

4. Wrong Answer?

- Quitar el freopen,

- no mezclar cin con scanf

- Ver si hay que imprimir fin de linea

- Leer nuevamente el problema.

- Ver si es multiples casos, repetir el mismo caso varias veces.

- long long

- Posibles Casos:

* $n = 0, n = -1, n = 1, n = 2^{31} - 1$ o $n = -2^{31}$

* La lista esta vacía o con un solo elemento

* n is even, n is odd

* El Grafo esta vacio o contiene un solo vertice

* El Grafo es un multigrafo (lazo o multiple aristas)

* El Poligono es convexo o no

- Hay condición inicial para los casos pequeños

- Estas utilizando el algoritmo correcto

- Explique su solución a alguien

- Puede que usted (o alguien más) debiera reescribir la solución?

5. Run-Time Error?

- Verificar el tamaño de los arreglos

- División por 0

- Verificar map, set, queue,...