Final algoritmos

Complejidad de tiempo

A screenshot of a cell phone

Description automatically generated

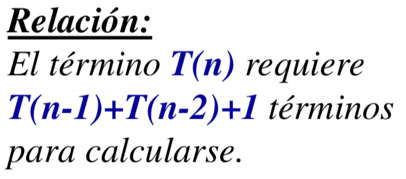
* O(1)
* O(logn)
  + Variable de control se multiplica o divide por un valor constante
* O(n)
  + Variable de control se incrementa o decrementa con valor constante
* O(nlogn)
* O(n^2)
* O(n^3)
* O(n^m) polinomeal
* O(m^n) exponencial
* O(n!) factorial

Recursividad

* Lineal: si es una sola llamada recursiva con incrementos o decrementos
* Logaritmico: si es una sola llamada con multiplicaciones o divisiones
* Exponencial: si hay más de una llamada recursiva

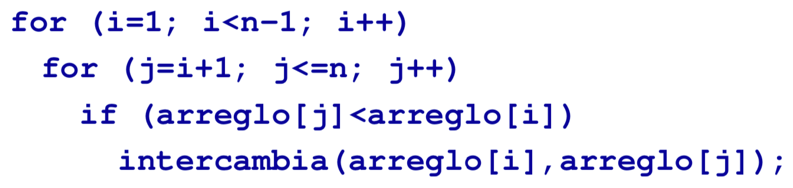
Fibonacci

* Iterativo: O(n)
* Recursivo: O(2^n/2)



Sort por intercambio

* Complejidad: O(n^2)



Multiplicación de matrices

* O(n^3)

A screenshot of a cell phone

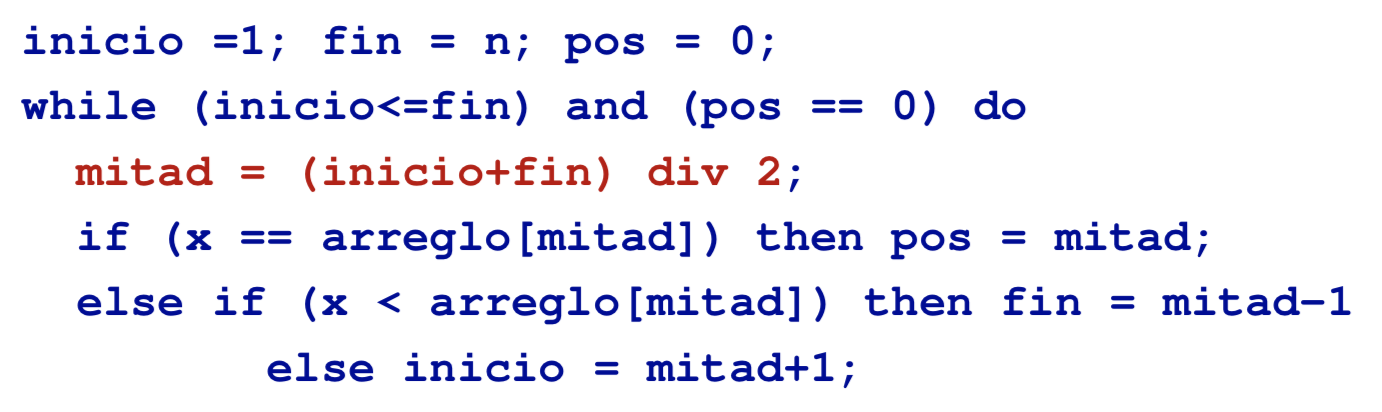
Description automatically generated

**Divide y Vencerás**

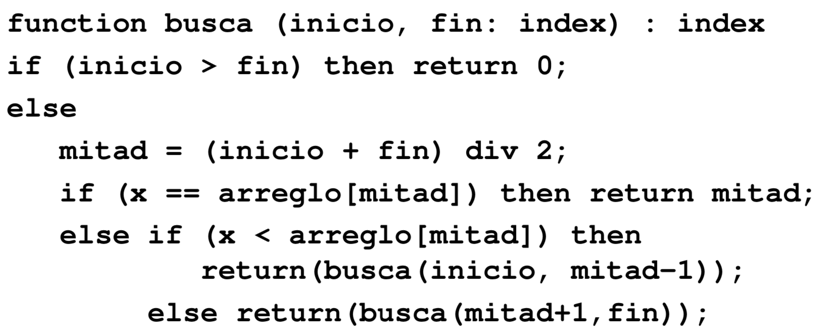
* Top-down, recursivo

Búsqueda Binaria

* O(logn)



Recursiva:



Merge Sort

* Divide en dos, ordena subarreglos y junta
* No es every-case por el une.
* Peor caso/caso promedio: O(nlogn)

A close up of a sign

Description automatically generated

A screenshot of a cell phone

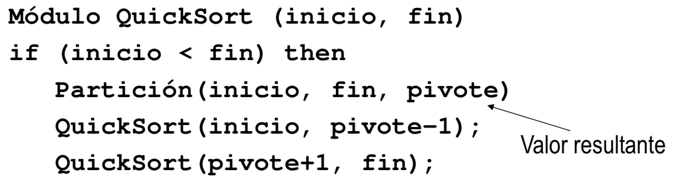
Description automatically generatedA screenshot of a cell phone

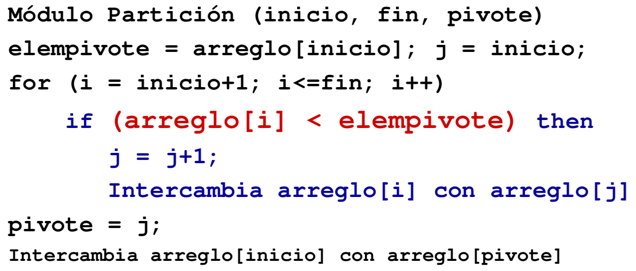
Description automatically generated

Quick Sort

* Se toma elemento pivote, divide en mayores y menores
* No es every-case por que la partición depende del pivote
* Peor caso: O(n^2), caso promedio: O(nlogn)

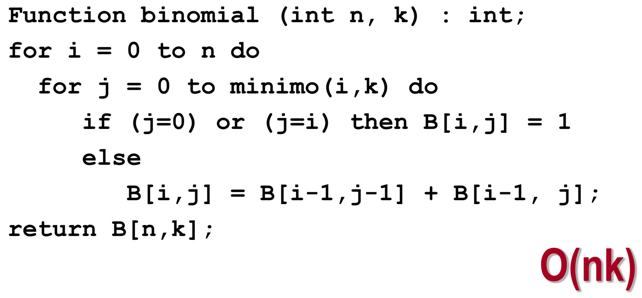
A picture containing object, clock

Description automatically generated



Programación dinámica

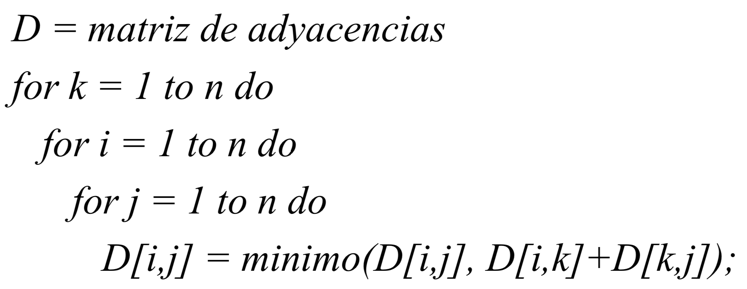
Coeficiente Binomial



Camino Más Corto

* Principio de optimalidad: Solución óptima contiene las soluciones óptimas de las subinstancias

Floyd



A screenshot of a cell phone

Description automatically generated

Multiplicación encadenada de matrices

* Siendo una matriz de nxm multiplicada por mxp, el calculo de un elemento requiere m multiplicaciones.
* Toda la matriz requiere mxnxp multiplicaciones escalares.
* Godbole O(n^3)
* A[7,8], dimensiones d6xd7, multiplicación de la matriz 7x8
* A = renglones de la primera, C = cantidad de columnas de la seg, B = renglones de la seg o columnas de la prim

A screenshot of a cell phone

Description automatically generatedA close up of a map

Description automatically generated

ABB Óptimo

A screenshot of a cell phone

Description automatically generated

* Gilbert and Moore

A picture containing text, map

Description automatically generated A screenshot of a cell phone

Description automatically generated

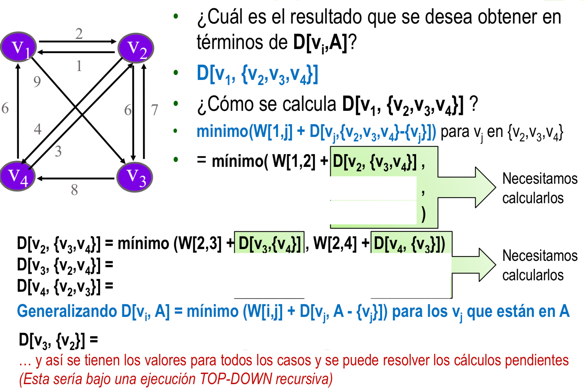
A screenshot of a cell phone

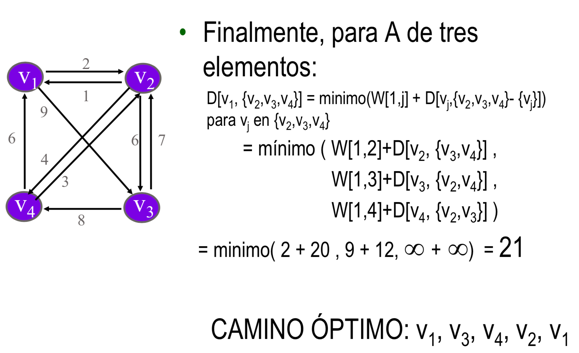
Description automatically generated

Problema del Viajero

* O(n^2\*2^n)

A picture containing object

Description automatically generated 

 A screenshot of a cell phone

Description automatically generated

Voraces

PRIM O(n^2) siendo n el número de vértices, muchos arcos (muy denso)

A close up of a watch

Description automatically generated

Kruskal O(n^2logn) ó (mlogm) siendo n la cantidad de vértices o m la cantidad de arcos

\*Pocos arcos

A close up of a clock

Description automatically generated

Dijkstra

O(n^2)

A screenshot of a cell phone

Description automatically generatedA close up of a clock

Description automatically generatedA close up of a clock

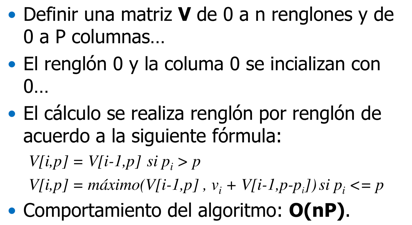
Description automatically generated

A close up of text on a white background

Description automatically generated

Problema de la Mochila

* Voraz solo cuando los objetos se pueden fraccionar
* Mayor valor por unidad de peso, O(nlogn).
* Programación dinámica O(nP), divide y venceras(2^n)



**Backtracking**

Tablero de reinas

Sum of Subsets

A screenshot of a cell phone

Description automatically generated

Coloreado de Grafos

* Cada nivel es un vértice, cada nodo es un color
* N hijos, n siendo la cantidad de colores
* X+1 niveles, siendo X la cantidad de estados

Ciclo hamiltoneano

* Cada nivel se incluye un vértice menos, raíz es el inicial

Problema de la Mochila O(2^n)

* Ordenar por valor/peso
* VP = Valor de los que caben + (Maximo – pesos que caben)\*(valor/peso)del sig

**Branch and Bound**

* Bread-first considerando best-first

A screenshot of a cell phone

Description automatically generated

Problema de la Mochila

* Ordenar por valor/peso, sacar VP, expandir el best-first

Problema del viajero

* Tendrá n-1 niveles, n es el número de nodos
* Sacar el costo posible, expandir el menor
  + Primer nodo: menor de cada renglón
  + Sig nodos: del último a todos menos del que viene
  + De todos a todos menos el útlimo