

Búsqueda exhaustiva

Metodologías de solución de problemas computacionales

Entendemos por metodología/técnica/paradigma a un esquema o “receta”, genérico y bien estructurado, para la solución de problemas que cumplen ciertas características.

A la luz de esta definición, durante este curso vamos a cubrir las siguientes metodologías:

- Búsqueda exhaustiva (brute-force search)
- Algoritmos voraces (greedy algorithms)
- Divide y vencerás (divide and conquer)
- Programación dinámica (dynamic programming)

+ Grafos

Búsqueda exhaustiva

Como su nombre lo indica, esta metodología consiste en validar sistemáticamente todos los posibles candidatos para la solución de un problema con el fin de encontrar el que cumple con un criterio dado.

Tiene como ventaja que generalmente es sencilla de implementar y, siempre que exista, encuentra una solución.

Tiene como desventaja que su eficiencia suele ser linealmente proporcional al número de soluciones candidatas, el cual suele ser polinomial o exponencialmente proporcional al tamaño del problema.

¿Qué algoritmo de los que hemos visto hasta el momento en el curso podría considerarse como búsqueda exhaustiva?

R/. Las soluciones 1 y 2 al problema de “Street Numbers”

```
read N
for i=2 to N-1:
    sumaIzq = 0
    for j=1 to i-1:
        sumaIzq += j
    sumaDer = 0
    for k=i+1 to N:
        sumaDer += k
    if sumaIzq = sumaDer:
        print i
        exit
print 'no hay'
```

```
read N
for i=2 to N-1:
    sumaIzq = i(i-1)/2
    sumaDer = (N(N+1)-i(i+1))/2
    if sumaIzq = sumaDer:
        print i
        exit
print 'no hay'
```

Búsqueda exhaustiva

Dado un problema P , el esquema general de la búsqueda exhaustiva se puede expresar como:

```
c = first(P) //Genera el primer candidato
while c ≠ ∅{
    if valid(P, c) then process(P, c)
    c = next(P, c) //Genera el siguiente candidato
}
```

valid(*P*, *c*) verifica si el candidato cumple con las condiciones del problema. En ocasiones la función *next* es tal que dicha verificación no es necesaria.

process(*P*, *c*) dependerá de lo que exija el problema.

Donde podemos ver este esquema en la solución 2 de “Street numbers”?

```
read N
i = 2
while i ≤ N-1:
    if true:
        process(i)
        i = i + 1
print 'no hay'

process(c):
    sumaIzq = i(i-1)/2
    sumaDer = (N(N+1)-i(i+1))/2
    if sumaIzq = sumaDer:
        print i
        exit
```