

Problema 2

Problema de la subsecuencia de suma máxima

- Disponemos de una secuencia de enteros (posiblemente negativos): $a_1, a_2, a_3, \dots, a_n$
- Encontrar la subsecuencia a_i, \dots, a_j tal que la suma $\sum_{k=i}^j a_k$ sea máxima
- Supondremos que la subsecuencia nula tiene un valor de suma igual a cero
 - Ello implica que si todos los números son negativos, la subsecuencia de suma máxima es la subsecuencia nula

Se pretende estudiar tres algoritmos diferentes para resolver este problema y analizar su eficiencia

Algoritmo 1: Fuerza Bruta

Mediante dos bucles anidados probamos con todas las posibles subsecuencias

- el primer bucle recorre todos los posibles comienzos de subsecuencia (para i desde 1 hasta n)
- el segundo bucle recorre todos los posibles finales de subsecuencia (para j desde i hasta n)

Mediante un tercer bucle calculamos la suma de la subsecuencia elegida

Almacenamos la solución más grande obtenida hasta el momento

- Si encontramos una solución mayor, la almacenamos en lugar de la última que hubiésemos encontrado

Diseño del algoritmo 1

```

método estático subsecuenciaSumaMaxima
  (entero[1..n] a)
  retorna subseq:tabla[1..2] de entero
  var
1     entero max:=0 // máximo hasta el momento
2     entero sumaActual
3     entero secIni:=0, secFin:=0 //solucion
  fvar

```

Diseño del algoritmo 1 (cont.)

```

4     para i desde 1 hasta n hacer
5         para j desde i hasta n hacer
6             sumaActual:=0
7             para k desde i hasta j hacer
8                 sumaActual:=sumaActual+a[k]
9             fpara
10            si sumaActual>max entonces
11                max:=sumaActual
12                secIni:=i
13                secFin:=j
14            fsi
15        fpara
16    retorna subseq:={secIni,secFin}
fmétodo

```

Algoritmo 2: Optimización sobre el anterior



Podemos darnos cuenta de que si ya hemos calculado la suma (s) de la subsecuencia a_i, \dots, a_j ,

- para obtener la suma de la siguiente subsecuencia a_i, \dots, a_{j+1} basta sumar $s+a_{j+1}$

Diseño del algoritmo 2



```
4  para i desde 1 hasta n hacer
5      sumaActual:=0
6      para j desde i hasta n hacer
7          sumaActual:=sumaActual+a[j]
8          si sumaActual>max entonces
9              max:=sumaActual
10             secIni:=i
11             secFin:=j
           fsi
       fpara
   fpara
12  retorna subseq:={secIni,secFin}
fmétodo
```

Algoritmo 3: Algoritmo de un solo bucle

El algoritmo se basa en las siguientes observaciones:

- Si una subsecuencia a_i, \dots, a_j tiene suma negativa, ninguna subsecuencia $a_i, \dots, a_j, \dots, a_q$ es de suma máxima
- Las subsecuencias que bordean a la subsecuencia máxima tienen suma cero o negativa
- Para cada valor de i , si la subsecuencia a_i, \dots, a_j es la primera de suma negativa, ninguna subsecuencia a_p, \dots, a_q siendo $i \leq p \leq j$ y $p \leq q$ es de suma máxima, o su suma es igual a la de otra secuencia ya considerada

Esto implica que en el cálculo de secuencias, cuando se detecta una secuencia negativa podemos avanzar i hasta $j+1$

Diseño del algoritmo 3

```
4   i:=1; j:=1; sumaActual:=0
5   mientras j<=n hacer
6       sumaActual:=sumaActual+a[j]
7       si sumaActual>max entonces
8           max:=sumaActual
9           secIni:=i; secFin:=j
       sino
10          si sumaActual<0 entonces
11              i:=j+1
12              sumaActual:=0
          fsi
      fsi
13   j++
14   fmientras
14   retorna subseq:={secIni,secFin}
```