

Exercici 7: Algorismes de cerca: cerca lineal.

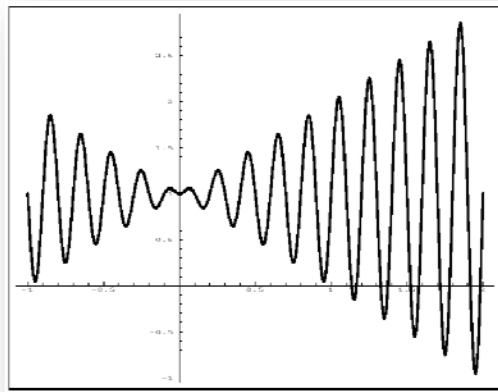
Lliurament:

UN ÚNIC FITXER (exercici7.py) QUE CONTINGUI EL CONJUNT DE FUNCIONS QUE S'HAN IMPLEMENTAT.

Criteris d'avaluació (per ordre d'importància):

- 1.- El programa ha de donar un resultat correcte: 50% de la nota.
- 2.- Ús adequat del llenguatge (fer servir if/while correctament, fer servir la col·lecció adequada, etc.): 30% de la nota.
- 3.- Bon estil de programació (fer una interfase d'usuari adequada, comentaris, etc.): 20% de la nota.

En aquest primer exercici de cerca ens concentrarem en el problema de cercar el màxim d'una funció en un determinat interval. Per exemple, considereu la funció $y = x \sin(10\pi x) + 1.0$ a l'interval $[-1.0, 2.0]$:



El problema que ens plantegem és trobar què val el màxim d'aquesta funció dins d'aquest interval i per quin valor de x el pren.

La primera aproximació que prendrem és la cerca per força bruta: generar **tots** els valors i determinar quin és el màxim. De totes maneres, per fer això podem usar certes funcionalitats de Python que ens faran el programa més elegant.

Una d'aquestes funcionalitats és la comanda `yield`. Aquesta comanda permet manegar de manera més eficient els valors que van sortint d'un iterador. Per entendre el seu funcionament, executeu aquest programa i observeu el seu comportament quan hi entreu un *string* qualsevol:

```
def all_perms(str):
    if len(str) <= 1:
        yield str
    else:
        for perm in all_perms(str[1:]):
            for i in range(len(perm)+1):
                yield perm[:i] + str[0:1] + perm[i:]
for i in all_perms("abcd"): print i
```

Aquesta comanda (que està en lloc del que podrien ser els `return` de la funció) ens ha creat un iterador, que genera elements només quan els necessitem.

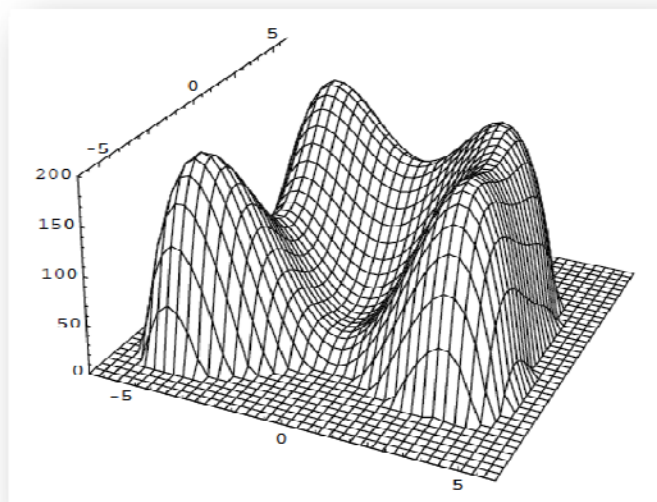
Per desenvolupar el programa (**exercici7.py**) aneu seguint els següents passos:

- Feu una funció **func1d(x)** que prengui el valor de x com a paràmetre i retorni el valor de la funció.
- Feu una funció **frange1d(start, end, inc)** que implenti la mateixa funcionalitat que té la instrucció **range** pels enters però pels reals. Feu que aquesta funció torni els valors amb un **yield** enlloc de retornar una llista.
- Feu una funció **search1d()** que retorni amb un **print** les informacions següents (en el mateix ordre):
 - Quin és el valor màxim de la funció a l'interval $[-1,2]$ quan mostregem els valors de x cada 0.01 unitats ($inc=0.01$), per quin valor de x es dona, quantes avaluacions de la funció heu fet i quan ha trigat.
 - Quin és el valor màxim de la funció a l'interval $[-1,2]$ quan mostregem els valors de x cada 0.0001 unitats ($inc=0.0001$), per quin valor de x es dona, quantes avaluacions de la funció heu fet i quan ha trigat.
 - Quin és el valor màxim de la funció a l'interval $[-1,2]$ quan mostregem els valors de x cada 0.000001 unitats ($inc=0.000001$), per quin valor de x es dona, quantes avaluacions de la funció heu fet i quan ha trigat.

Observacio:

El valor màxim de la funció està al voltant del valor 2,85 i es dona per una x al voltant del valor 1,85.

Reescriuiu les funcions anteriors (i creeu les funcions **func2d**, **frange2d**, **search2d**) per la funció $f(x, y) = 200 - (x^2 + y - 11)^2 - (x + y^2 - 7)^2$ a l'interval $-6 \leq x, y \leq +6$.



Observacions:

Si mostregeu (x,y) amb passos molt petits això pot arribar a trigar molt. Comproveu fins on podeu mostrejar per poder donar un resultat en menys de 10 segons.

Aquesta funció té quatre màxims amb el mateix valor al voltant dels punts $(3.58, -1.86)$, $(3.0, 2.0)$, $(-2.815, 3.125)$ i $(-3.78, -3.28)$.