

Práctica 1

1.1.1 Consideremos el problema

$$\begin{array}{ll} \text{mín } f(x) \\ \text{s.a. } x \in \Omega \end{array}$$

- a. Encontrar ejemplos donde todos los puntos de Ω sean minimizadores locales pero $f(x) \neq f(y)$ para $x \neq y$.
- b. Encontrar ejemplos de conjuntos Ω y funciones f que tengan varios minimizadores locales y globales.

1.1.2 Mostrar con ejemplos qué pasa cuando se eliminan de las hipótesis del teorema de Bolzano-Weierstrass las condiciones de continuidad o compacidad.

1.1.3 Probar que si f es continua en $\mathbb{R}^n$ y $\lim_{\|x\| \rightarrow +\infty} f(x) = +\infty$ entonces f tiene un minimizador global en $\mathbb{R}^n$.

1.1.4 Probar que si f es continua en $\mathbb{R}^n$ y para algún $x_0 \in \mathbb{R}^n$ el conjunto de nivel $\{x \in \mathbb{R}^n | f(x) \leq f(x_0)\}$ es acotado, entonces f tiene un minimizador global en $\mathbb{R}^n$.

Algoritmos y complejidad

1.3.1 a. Decidir si son verdaderas o falsas las siguientes afirmaciones y justificar:

- I. n es $O(n)$
 - II. $3n^2 + 7n + 4$ es $O(n^2)$
 - III. n^i es $O(n^j)$ si $i < j$
 - IV. $n \log_2(n)$ es $O(n^2)$
- b. Mostrar que $n!$ no es $O(r^n)$, cualquiera sea el entero positivo r . Concluir que una complejidad de $O(n!)$ es “peor” que exponencial de cualquier base.
 - c. Mostrar que $n \log n$ es $O(\log(n!))$ y $\log(n!)$ es $O(n \log n)$.

1.3.2 Decir cual es la complejidad de los siguientes algoritmos:

a. Dadas dos variables x, y :PASO 1: poner $aux = x$ PASO 2: poner $x = y$ PASO 3: poner $y = aux$ b. Dado un vector v , su dimensión d y un elemento e :PASO 1: poner $i = 1$ PASO 2: mientras $i < d$ y $e \neq v[i]$ hacerPASO 3: $i = i + 1$ PASO 4: si $e = v[i]$ imprimir i

PASO 5: sino imprimir ‘‘No se encontró’’

c. Dado un número natural n :función $Factorial(n)$ si $n = 0$ devolver 1sino devolver $(n \times Factorial(n - 1))$

fin función

d. Dado un vector v , y su dimensión d :PASO 1: para i desde 1 hasta d hacerPASO 2: poner $min = i$ PASO 3: para j desde i hasta d hacerPASO 4: si $v[j] < v[min]$ poner $min = j$ PASO 5: poner $aux = v[i]$ PASO 6: poner $v[i] = v[min]$ PASO 7: poner $v[min] = aux$

1.3.3 a. Escribir algoritmos para realizar suma, resta y multiplicación de matrices. Determinar la complejidad y expresarla en función de la dimensión de las matrices.

b. Escribir un algoritmo que dada una matriz A y un número n calcule A^n . Determinar la complejidad del mismo.c. Escribir y determinar la complejidad del algoritmo de Gauss (triangulación) para resolver un sistema de n ecuaciones lineales con n incógnitas.