

UNIVERSIDAD NACIONAL MAYOR DE SAN MARCOS
FACULTAD DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA
INE018 MATEMÁTICA COMPUTACIONAL

Solucionario del examen final
2024-1

Indicaciones generales:

- Duración: 120 minutos.
- Materiales o equipos a utilizar: 2 hojas A4 con apuntes de clase (físicos).
- No está permitido el uso de ningún material o equipo electrónico adicional al indicado (no celulares, no tablets, no libros).
- **La presentación, la ortografía y la gramática de los trabajos influirán en la calificación.**

Puntaje total: 20 puntos.

Pregunta 1. (5 puntos)

Una medida estadística común es la **desviación estándar**, la cual provee un indicador de qué tanto los valores en la distribución x_1, \dots, x_n difieren de la media. Matemáticamente, la desviación estándar σ es expresada como

$$\sigma = \frac{\sqrt{\sum_{i=1}^n (\bar{x} - x_i)^2}}{n}.$$

Escriba una función `long double DesviacionEstandar(vector<long double>& datos)` que retorne la desviación estándar de la distribución de datos.

```
long double Media(const vector<long double>& datos) {
    long double suma = 0.0L;
    for (long double dato : datos) suma += dato;
    return suma / datos.size();
}

long double DesviacionEstandar(const vector<long double>& datos) {
    long double media = Media(datos);
    long double suma = 0.0L;
    for (long double dato : datos) {
        long double delta = media - dato;
        suma += delta * delta;
    }
    return sqrt(suma) / datos.size();
}
```

Pregunta 2. (5 puntos)

Implemente una función `bool SonConjuntosDisjuntos(set<int> S, set<int> T)` que reciba dos conjuntos S y T , y retorne si $S \cap T = \emptyset$.

```
bool SonConjuntosDisjuntos(const set<int>& S, const set<int>& T) {
    for (int s : S) {
        if (T.contains(s)) {
            return false;
        }
    }
    return true;
}
```

Pregunta 3. (5 puntos)

Escriba una función recursiva que halle el máximo elemento de un vector de tamaño $n \geq 1$.

```
int Maximo(const vector<int>& v, int n) {
    if (n == 1) return v[0];
    return max(v[n - 1], Maximo(v, n - 1));
}
```

Pregunta 4. (5 puntos)

Demuestre que n^2 está en $\mathcal{O}(2^n)$.

Prueba. Computamos algunos valores iniciales de las funciones n^2 y 2^n :

n	n^2	2^n
1	1	2
2	4	4
3	9	8
4	16	16
5	25	32

Observamos que para $n = 4$ coinciden, y para $n > 4$, 2^n crece más rápidamente que n^2 . Formulamos entonces la hipótesis de que $n^2 \leq 2^n$ para $n \geq 4$. Utilizamos inducción matemática para probar esta afirmación. Para $n = 4$, claramente $4^2 \leq 2^4$. Supongamos que la hipótesis se cumple para un $n \geq 4$, es decir, que $n^2 \leq 2^n$. Debemos demostrar que $(n + 1)^2 \leq 2^{n+1}$. Para $n \geq 4$, sabemos que $2n + 1 \leq n^2$, ya que $(n - 1)^2 - 2 \geq 0$. Utilizando esta desigualdad y la hipótesis inductiva, conseguimos lo siguiente:

$$(n + 1)^2 = n^2 + 2n + 1 \leq 2n^2 \leq 2^{n+1}.$$

Concluimos que $n^2 \leq 2^n$ para todo $n \geq 4$. Así, $n^2 \in \mathcal{O}(2^n)$. □

Profesor del curso: Manuel Loaiza Vasquez.

Lima, 13 de julio de 2024.