

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

Solucionario del examen sustitutorio
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. (7 puntos)

Implemente dos pilas en un vector de tamaño n de tal manera que ninguna de las pilas se desborde salvo que el número total de elementos entre ambas pilas sea n . Las operaciones Push y Pop deben correr en $\mathcal{O}(1)$ en tiempo.

```
/*
 * n[0]: Tamaño fijo del vector v.
 * n[1]: Número de elementos en la primera pila. Inicialmente cero.
 * n[2]: Número de elementos en la segunda pila. Inicialmente cero.
 * No olvide inicializar este vector antes de manipular las pilas.
 */
vector<int> n;

/*
 * Vector de tamaño fijo que contendrá los elementos de las dos pilas.
 * No olvide redimensionarlo una vez inicializado el vector n.
 */
vector<int> v;

// Inserta x en la i-ésima pila.
void Push(int i, int x) {
    if (n[1] + n[2] == n[0]) {
```

```

        // ERROR: Memoria insuficiente.
    }
    if (i == 1) {
        v[n[1]] = x;
    } else {
        v[n[0] - 1 - n[2]] = x;
    }
    n[i]++;
}

// Elimina el valor en el tope de la i-ésima pila.
void Pop(int i) {
    if (n[i] == 0) {
        // ERROR: No se puede desapilar de una pila vacía.
    }
    n[i]--;
}

```

Pregunta 2. (7 puntos)

2.1. Escriba una función recursiva `int SumaDeDigitos(int n)` que tome un entero no negativo y retorne su suma de dígitos. Por ejemplo, la llamada `SumaDeDigitos(1729)` debería retornar $1 + 7 + 2 + 9$, lo cual es 19.

```

int SumaDeDigitos(int n) {
    if (n < 10) return n;
    return n % 10 + SumaDeDigitos(n / 10);
}

```

2.2. La **raíz digital** de un entero no negativo n está definida como el resultado de sumar los dígitos repetidas veces hasta que solo quede un dígito. Por ejemplo, la raíz digital de 1729 puede ser calculada usando los siguientes pasos: $1 + 7 + 2 + 9 \rightarrow 19$, $1 + 9 \rightarrow 10$ y $1 + 0 \rightarrow 1$. Debido a que la suma al final del tercer paso es el dígito 1, ese valor es la raíz digital. Escriba una función recursiva `int RaizDigital(int n)` que retorne la raíz digital de su argumento.

```

int RaizDigital(int n) {
    if (n < 10) return n;
    return RaizDigital(SumaDeDigitos(n));
}

```

Pregunta 3. (6 puntos)

Sean f y g dos funciones eventualmente no negativas definidas sobre los naturales. Demuestre que $\max(f(n), g(n))$ está en $\Theta(f(n) + g(n))$.

Prueba. Como f y g son funciones eventualmente no negativas, existen constantes positivas n_f y n_g tales que $f(n) \geq 0$ para $n \geq n_f$ y $g(n) \geq 0$ para $n \geq n_g$. Queremos encontrar constantes $c_1 > 0$, $c_2 > 0$ y $n_0 > 0$ tales que $0 \leq c_1(f(n) + g(n)) \leq$

$\max(f(n), g(n)) \leq c_2(f(n) + g(n))$ para todo $n \geq n_0$. Afirmando que $c_1 = 1/2, c_2 = 1$ y $n_0 = \max(n_f, n_g)$ cumplen lo requerido. Sea $n \geq n_0$ un número natural fijo pero arbitrario. Sabemos que la mayor de las imágenes domina a cualquiera de las funciones; es decir, $0 \leq f(n) \leq \max(f(n), g(n))$ y $0 \leq g(n) \leq \max(f(n), g(n))$. Combinando estas desigualdades conseguimos $0 \leq (f(n) + g(n))/2 \leq \max(f(n), g(n))$. Asimismo, el máximo de las imágenes debe coincidir con alguna de las imágenes para cada punto en el dominio. Sin pérdida de generalidad, supongamos que coincide con la imagen de f en el punto fijado $n \geq n_0$. Luego, $\max(f(n), g(n)) = f(n) \leq f(n) + g(n)$. En caso no coincida con f en dicho punto, necesariamente coincide con g y obtendríamos $\max(f(n), g(n)) = g(n) \leq f(n) + g(n)$. Así, $\max(f(n), g(n)) \leq f(n) + g(n)$. \square

Profesor del curso: Manuel Loaiza Vasquez.

Lima, 20 de julio de 2024.