

Cálculo de $T(n)$

Colectivo Estructuras de Datos y Algoritmos

Marzo 2021

1. Sean A y B dos algoritmos cuyas complejidades temporales están representadas por las funciones dadas. ¿Para qué valores de n , el algoritmo A ejecuta más rápido que el algoritmo B en una misma arquitectura?

- a) \mathbf{A} : $T_a(n) = 8n^2$, \mathbf{B} : $T_b(n) = 64n$
b) \mathbf{A} : $T_a(n) = 100n^2$, \mathbf{B} : $T_b(n) = 2^n$

2. Determine la función de complejidad temporal $T(n)$ de los siguientes algoritmos, siendo n la longitud de la lista A . Analice para qué características de la entrada, $T(n)$ alcanza su mayor y menor valor.

a)

```
1  bool IsSorted(int[] A){
2      for(int i = 1; i < A.Length(); i++){
3          if(A[i] < A[i-1])
4              return false;
5      }
6      return true;
7  }
```

b)

```
1  void Sort(int[] A){
2      for(int i = 0; i < A.Length(); i++)
3          for(int j = i + 1; i < A.Length(); j++)
4              if(A[i] > A[j]){
5                  tmp = A[i];
6                  A[i] = A[j];
7                  A[j] = tmp;
8              }
9  }
```

c)

```
1  int Find(int[] A, int key) {
2      for(int i = 0; i < A.Length(); i++)
3          if(A[i] == key) return i;
4      return -1;
5  }
```

3. Para los siguientes algoritmos, ¿cuál es el tamaño de la entrada? Determine la función de complejidad temporal $T(n)$ en cada caso.

a)

```
1  int IsPrime(int x) {
2      if(x <= 1) return false;
3      int d = 2;
4      while(x % d != 0 && d*d <= x) d++;
5      return d*d > x;
6  }
```

b)

```
1  int Fibonacci(int x) {
2      if (x == 0) return 0;
3      if (x == 1) return 1;
4
5      int A = 0, B = 1;
6      for (int i = 2; i <= x; i++) {
7          int C = A + B;
8          A = B;
9          B = C;
10     }
11     return B;
12 }
```

4. Proponga un algoritmo que para un x fijo, calcule el valor de x^n . Determine la función de complejidad temporal $T(n)$ de su algoritmo.

NOTA: Solo por esta vez, los códigos presentados en la asignatura están escritos, por conveniencia, en C#. De manera usual se hará uso de *pseudocódigo* para presentar y analizar las soluciones de los ejercicios.