

Algoritmos y Estructuras de Datos Avanzadas

Magíster en Ciencias de la Computación

Gilberto Gutiérrez

Universidad del Bío-Bío
Departamento de Ciencias de la Computación y TI

April 17, 2020

Indice

- 1 Coordinación del curso
- 2 Programa
 - Descripción
 - Resultados de aprendizaje
 - Contenidos
 - Metodología
 - Evaluación
 - Bibliografía
- 3 RA 1

Coordinación del curso

- Plataforma adecca
- Zoom para video conferencia
- Clases por video conferencia (40-50 minutos) 1 vez por semana (viernes 14:10-15:00)
- Trabajos individuales para complementar las clases
- Una sesión de consultas (dudas) por video conferencia 1 vez por semana (horario flexible)
- Consultas por email (los que sean necesarios)

Descripción

Descripción

- **Obligatorio, Teórico-práctica**
- Profundizar: Estructuras de datos y Algoritmos
- Diseñar y analizar algoritmos: Determinístico, Probabilísticos, Paralelos y Aproximados
- Algoritmos y estructuras de datos en dominios particulares: Geometría computacional, búsqueda en texto, grafos
- Complejidad Computacional.

Competencias

- Diseñar y evaluar algoritmos y estructuras de datos
- Analizar un problema y determinar su dificultad y clasificarlo de acuerdo a su complejidad computacional

Resultados de aprendizaje

- ① RA 1: Diseña algoritmos y realiza análisis temporal y espacial para determinar su complejidad.
- ② RA 2: Genera soluciones algorítmicas sobre modelos de computación secuencial, paralela o distribuida
- ③ RA 3: Utiliza estructuras de datos complejas y diseña algoritmos eficientes sobre tales estructuras
- ④ RA 4: Determina la complejidad computacional de un problema para establecer si pertenece a la clase P, NP, NP-Completo u otra.

Contenidos

RA 1

- Operaciones relevantes
- Escenarios de análisis (mejor, peor y caso promedio)
- Recursos consumidos por el algoritmo
- Funciones de costo para algoritmos
- Cotas para funciones de costo de algoritmos (O , Ω , Θ , o , ω)
- Técnicas de diseño de Algoritmos (dividir para reinar, algoritmos voraces, backtracking, programación dinámica, ramificación y poda)

Contenidos

RA 2

- Modelos de computación secuencial, paralelo y distribuido
- Algoritmos paralelos.
- Algoritmos probabilísticos
- Algoritmos aproximados
- Algoritmos distribuidos

Contenidos

RA 3

- Estructuras de datos complejas (espaciales, árboles B, Hashing, Skip-Lists, Kd-tree, Range-tree, etc.)
- Algoritmos en texto, en grafos y geométricos.

RA 4

- Clases de problemas P , NP , NP -Completo, NP -Hard.
- Reducción de problemas
- Algoritmos heurísticos, aproximados y probabilísticos

Metodología

- Clases expositivas interactivas.
- Análisis de artículos.
- Trabajo individual.
- Trabajo colaborativo

Evaluación

- 3 Tareas (30%)
- 1 Trabajo de Investigación (30%)
- 1 Certamen de fin de semestre (40%)

Bibliografía

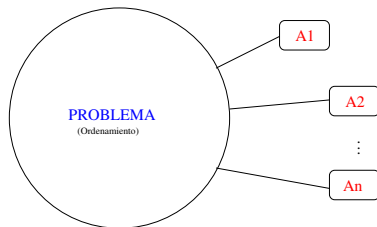
- Fundamental

- Cormen, T., Leiserson, C., Rivest, R., Stein, C. (2009), Introduction to Algorithm, The MIT Press, 3 edition.
- Knuth, D. (1998), The Art of Computer Programming, Addison Wesley.
- Samet, H. (2005), Foundations of Multidimensional and Metric Data Structures, The Morgan Kaufmann Series in Computer Graphics, 1 edition.

- Complementaria

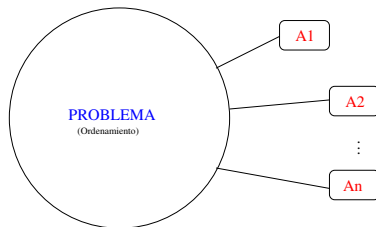
- Aho, A., Hopcroft, J., Ullam, J. (1998), Estructuras de Datos y Algoritmos, Addison Wesley.
- Preparata, F., Shamos, M. (1985), Computational Geometry, an Introduction, Springer.
- Navarro, G., Raffinos, M. (2002), Flexible Matching in Strings, Cambridge University Press.
- Paradimitriou, C. (1998), Combinatorial Optimization: Algorithms and Complexity, Dover.
- Knuth, R., Patashnik, O. (1990), Concrete Mathematics, Addison Wesley.

Motivación



Preguntas

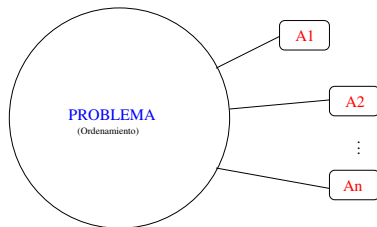
Motivación



Preguntas

1. ¿ A_1 Es mejor que A_2 ?

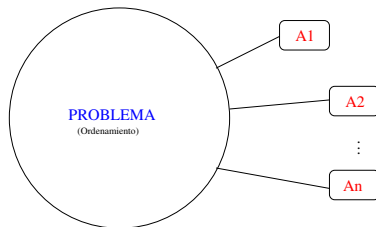
Motivación



Preguntas

1. ¿ A_1 Es mejor que A_2 ?
2. ¿Cuál es el mejor algoritmo?

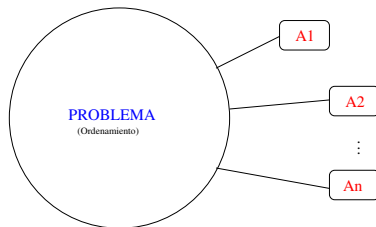
Motivación



Preguntas

1. ¿ A_1 Es mejor que A_2 ?
2. ¿Cuál es el mejor algoritmo?
3. ¿En qué escenarios A_2 es mejor que A_1 ?

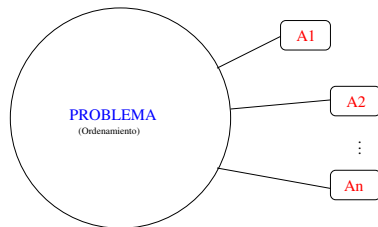
Motivación



Preguntas

1. ¿ A_1 Es mejor que A_2 ?
2. ¿Cuál es el mejor algoritmo?
3. ¿En qué escenarios A_2 es mejor que A_1 ?
4. ¿Habrà un algoritmo A_{n+1} mejor que los que se conocen?

Motivación



Preguntas

1. ¿ A_1 Es mejor que A_2 ?
2. ¿Cuál es el mejor algoritmo?
3. ¿En qué escenarios A_2 es mejor que A_1 ?
4. ¿Habrà un algoritmo A_{n+1} mejor que los que se conocen?
5. ¿Cuál es la cota inferior del problema?
6. ¿...?

Motivación

- Ej. Dado un arreglo A de tamaño n , ordenar los elementos de A de menor a mayor

Algorithm 1 SortBasico(A, n)

```
1: Input:  $A, n$ 
2: Output:  $A$  ordenado de menor a mayor
3: for  $i = 1$  to  $n - 1$  do
4:   for  $j = i + 1$  to  $n$  do
5:     if  $a[i] > a[j]$  then
6:       Intercambia( $A, i, j$ )
7:     end if
8:   end for
9: end for
```

Motivación

Algorithm 2 InsertSort(A, n)

```
1: Input:  $A, n$ 
2: Output:  $A$  ordenado de menor a mayor
3: for  $j = 1$  to  $n$  do
4:    $key = a[j]$ 
5:    $i = j - 1$ 
6:   while  $i \geq 0$  and  $a[i] > key$  do
7:      $a[i + 1] = a[i]$ 
8:      $i = i - 1$ 
9:   end while
10:   $a[i + 1] = key$ 
11: end for
```

Motivación

Algorithm 3 HeapSort(A, n)

```
1: Input:  $A, n$ 
2: Output:  $A$  ordenado de menor a mayor
3: BuildHeap( $A$ )
4: for  $i = n - 1$  to 0 do
5:   Intercambia( $A, 0, i$ )
6:   Heapify( $A, 0, i - 1$ )
7: end for
```

Motivación

Algorithm 4 BuildHeap(A, n)

```
1: Input:  $A, n$ 
2: Output: Un heap en  $A$ 
3: for  $i = n/2; i \geq 0; i --$  do
4:   Heapify( $A, i, n - 1$ )
5: end for
```

Motivación

Algorithm 5 Heapify(A, i, j)

```
1: Input:  $A, i, j$ 
2: Output: Heapify
3: if  $(2 * i + 1) \leq j$  then
4:   if  $(2 * i + 2) \leq j$  then
5:     if  $(a[2 * i + 2] \geq a[2 * i + 1])$  then
6:        $k = 2 * i + 2$ 
7:     else
8:        $k = 2 * i + 1$ 
9:     end if
10:  else
11:     $k = 2 * i + 1$ 
12:  end if
13:  if  $a[i] < a[k]$  then
14:    Intercambia( $A, i, k$ );
15:    Heapify( $A, k, j$ )
16:  end if
17: end if
```

¿ Que medimos ?

Recursos consumidos por el algoritmo

¿ Que medimos ?

Recursos consumidos por el algoritmo

- Tiempo

¿ Que medimos ?

Recursos consumidos por el algoritmo

- Tiempo
- Memoria (Almacenamiento)

Análisis del algoritmo (consumo de recursos)

¿ Que medimos ?

Recursos consumidos por el algoritmo

- Tiempo
- Memoria (Almacenamiento)

Análisis del algoritmo (consumo de recursos)

- Empírico

¿ Que medimos ?

Recursos consumidos por el algoritmo

- Tiempo
- Memoria (Almacenamiento)

Análisis del algoritmo (consumo de recursos)

- Empírico
- Teórico