

**PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ**  
**FACULTAD DE CIENCIAS E INGENIERÍA**

**ALGORITMOS AVANZADOS**

**Segundo Examen  
(Segundo Semestre 2024)**

Duración: 2h 50 min.

- **No puede utilizar apuntes, solo hojas sueltas en blanco.**
- En cada función el alumno deberá incluir, a modo de comentario, la forma de solución que utiliza para resolver el problema. De no incluirse dicho comentario, el alumno perderá el derecho a reclamo en esa pregunta.
- No puede emplear plantillas o funciones no vistas en los cursos de programación de la especialidad.
- Los programas deben ser desarrollados en el lenguaje C++. Si la implementación es diferente a la estrategia indicada o no la incluye, la pregunta no será corregida.
- Un programa que no muestre resultados coherentes y/o útiles será corregido sobre el 50% del puntaje asignado a dicha pregunta.
- Debe utilizar comentarios para explicar la lógica seguida en el programa elaborado. El orden será parte de la evaluación.
- Se utilizarán herramientas para la detección de plagios, por tal motivo si se encuentran soluciones similares, se anulará la evaluación a todos los implicados y se procederá con las medidas disciplinarias dispuestas por la FCI.
- **Solo está permitido acceder a la plataforma de PAIDEIA, cualquier tipo de navegación, búsqueda o uso de herramientas de comunicación se considera plagio por tal motivo se anulará la evaluación y se procederá con las medidas disciplinarias dispuestas por la FCI.**
- Para esta evaluación solo se permite el uso de las librerías **iostream, iomanip, climits, cmath, fstream, vector, list, map, algorithm, cstring**
- Su trabajo deberá ser subido a PAIDEIA.
- **Es obligatorio usar como compilador NetBeans.**
- Los archivos deben llevar como nombre su código de la siguiente forma `codigo_EX2_P#` (donde # representa el número de la pregunta a resolver)

---

**Programación en Ventanas de Tiempo**

**Planteamiento del problema**

Se tiene un conjunto de **n tareas**  $\{T_1, T_2, \dots, T_n\}$  cada una con:

- Una **duración**  $d_i$
- Una **ventana de tiempo**  $[s_i, e_i]$ , donde  $s_i$  es el inicio y  $e_i$  el final de la ventana en la que puede programarse  $T_i$
- Una **penalización por tardanza**  $w_i$  que se incurre si la tarea no se ejecuta dentro de su ventana.

El objetivo es **programar todas las tareas de manera que se minimice la penalización total**, cumpliendo las restricciones de las ventanas de tiempo.

Ejemplo:

Tarea ( $T_i$ )	Duración ( $d_i$ )	Ventana de tiempo ( $[s_i, e_i]$ )	Penalización ( $w_i$ )
$T_1$	2	[1, 4]	10
$T_2$	3	[3, 6]	15
$T_3$	1	[5, 7]	8
$T_4$	2	[2, 5]	12
$T_5$	4	[6, 10]	20
$T_6$	2	[8, 11]	7
$T_7$	1	[9, 10]	6
$T_8$	3	[1, 3]	18
$T_9$	2	[4, 7]	14
$T_{10}$	3	[6, 9]	16

**Paso 1: Ordenar tareas por  $e_i$  (ventana de término más temprana)**

Después de ordenar, tenemos:

1.  $T_8$ : [1,3]
2.  $T_1$ : [1,4]
3.  $T_4$ : [2,5]
4.  $T_2$ : [3,6]
5.  $T_9$ : [4,7]
6.  $T_3$ : [5,7]
7.  $T_{10}$ : [6,9]
8.  $T_5$ : [6,10]
9.  $T_7$ : [9,10]
10.  $T_6$ : [8,11]

**Paso 2: Asignación Greedy (priorizando tareas con menor  $e_i$ )**

1.  $T_8$ :
  - Duración 3, se programa de 1 a 3.
  - **Estado actual:**  $T_8$  programada.
2.  $T_1$ :
  - Duración 2, ventana [1,4] pero no hay espacio disponible.
  - Penalización: 10
3.  $T_4$ :
  - Duración 2, se programa de 4 a 5.
  - **Estado actual:**  $T_8, T_4$  programada.
4.  $T_2$ :
  - Duración 3, ventana [3,6] pero solo queda [6] (1 unidad disponible).
  - Penalización: 15.
5.  $T_9$ :
  - Duración 2, se programa de 6 a 7.
  - **Estado actual:**  $T_8, T_4, T_9$  programadas.
6.  $T_3$ :
  - Duración 1, ventana [5,7] pero no hay espacio disponible.
  - Penalización: 8.

7. **T<sub>10</sub>:**
- Duración 3, ventana [6,9] pero ya está ocupada.
  - Penalización: 16.
8. **T<sub>5</sub>:**
- Duración 4, ventana [6,10] pero ya está ocupada
  - Penalización: 20.
9. **T<sub>7</sub>:**
- Duración 1, se programa a las 9.
  - **Estado actual:** T<sub>8</sub>, T<sub>4</sub>, T<sub>9</sub>, T<sub>7</sub> programadas.
10. **T<sub>6</sub>:**
- Duración 2, puede programarse de 10 a 11.
  - **Estado actual:** T<sub>8</sub>, T<sub>4</sub>, T<sub>9</sub>, T<sub>7</sub>, T<sub>6</sub> programadas.

#### **Resultados finales**

- **Tareas programadas:** T<sub>8</sub>, T<sub>4</sub>, T<sub>9</sub>, T<sub>7</sub>, T<sub>6</sub>
- **Tareas no completadas:** T<sub>1</sub>, T<sub>2</sub>, T<sub>3</sub>, T<sub>10</sub>, T<sub>5</sub>.
- **Penalización total:** 10+15+8+16+20=69.

#### **Pregunta 1 (10 puntos)**

Desarrolle un algoritmo GRASP para dar solución al problema de programación en ventanas de tiempo. Para el desarrollo debe emplear como base el criterio Greddy detallado. La respuesta para este ejemplo, con un alfa=0.4 y 10,000 iteraciones es:

**Tareas programadas:** T<sub>8</sub>, T<sub>9</sub>, T<sub>3</sub>, T<sub>10</sub>, T<sub>7</sub>

**Penalización total:** 64

#### **Pregunta 2 (10 puntos)**

Desarrolle un algoritmo Genético para dar solución al problema de programación en ventanas de tiempo.

##### **Parámetros del algoritmo genético**

1. **Tamaño de población:** 10 cromosomas.
2. **Generaciones:** 1000.
3. **Probabilidad de selección o casamiento:** 0.8 (80%).
4. **Probabilidad de mutación:** 0.5 (50%).
5. **Selección de padres:** Método de la ruleta.

Una respuesta recurrente para este ejemplo es:

**Tareas programadas:** T<sub>5</sub>, T<sub>7</sub>, T<sub>8</sub>, T<sub>9</sub>

**Penalización total:** 68

Al finalizar el examen, comprima la carpeta de su proyecto empleando el programa Zip que viene por defecto en el Windows, **no se aceptarán los trabajos compactados con otros programas como RAR, WinRAR, 7zip o similares**. Luego súbalo a la tarea programa en Paideia para este examen.

Profesores del curso:

Manuel Tupia  
Rony Cueva

San Miguel, 10 de diciembre del 2024