

Tarea semanal 2.2: Backtracking

1. Objetivos

Resolver problemas mediante la técnica *Backtracking*.

2. Conocimientos previos

Backtracking.

3. Preguntas

- ¿Cómo implementa Backtracking la recorrida sobre el espacio de soluciones?
- ¿Qué diferencia a Backtracking de los algoritmos de “fuerza bruta”?
- ¿Qué es la función objetivo y cuándo corresponde tenerla?
- Si no se controla un predicado de poda, ¿se pierden soluciones?
- ¿Cómo se determina la performance (en general) de los algoritmos de Backtracking?

4. Problema

Una universidad está interesada en albergar un congreso en sus instalaciones. Ella cuenta con m salas de igual capacidad, y para el congreso están planificadas n charlas, cada una con una cantidad estimada de *asistentes*. Las capacidades de cada sala son suficientes para albergar a los potenciales asistentes de cualquiera de las charlas.

Se pretende asignar las salas a las charlas de manera tal que el total de asistentes sea máximo. Pueden quedar charlas sin ser desarrolladas.

Cada charla tiene definida el intervalo de tiempo en la que se desarrollaría. Además cada sala tiene un intervalo de disponibilidad. Cada intervalo, tanto de desarrollo de charlas como de disponibilidad de salas, se define por los atributos *inicio* y *fin*, que son conocidos. Estos valores son enteros no negativos. Una sala puede asignarse a una charla si su intervalo de disponibilidad contiene al intervalo de desarrollo de la charla. Tanto los intervalos de desarrollo de las charlas como los de disponibilidad de las salas pueden superponerse.

La asignación de una sala a una charla debe hacerse de forma exclusiva y para todo el intervalo de desarrollo de la charla. Por lo tanto, para que a dos charlas se les asigne la misma sala, esas charlas deben ser compatibles, lo cual significa que sus intervalos de desarrollo no se superponen.

Se consideran dos problemas:

max_asistentes :

- En este caso hay una sola sala (esto es, $m = 1$).
- El intervalo de disponibilidad de esa sala contiene al de desarrollo de cada una de las charlas.

max_asistentes_m : Es el problema general.

5. Entrega

La solución debe incluir lo siguiente:

1. Respuestas de las preguntas de la sección 3.
2. Solución en términos de Backtracking de los problemas de la sección 4.

Para cada uno de ellos:

- a) Formalice el problema: forma de la solución, restricciones explícitas e implícitas, función objetivo y predicados de poda en caso de que corresponda.
- b) Diseñe ejemplos, contemplando todas las situaciones posibles, y dibuje para cada uno de ellos el árbol de soluciones. Muestre, para cada nodo, qué restricciones se cumplen, cuál es el valor de la función objetivo y si se debe podar y por qué motivo.

La tupla de *max_asistentes* debe ser de largo variable, mientras que la de *max_asistentes_m* debe ser de largo fijo.

Se asume que existen los siguientes parámetros y estructuras:

- *n* charlas en un array llamado *charlas* con índices entre 0 y *n* − 1.
- *m* salas en un array llamado *salas* con índices entre 0 y *m* − 1.

Además, está definida la función *compatibles* para $i, j \in \{0, \dots, n - 1\}$:

$$\text{compatibles}(i, j) = \begin{cases} \text{true} & \text{si } \text{charlas}[i].\text{fin} \leq \text{charlas}[j].\text{inicio} \\ & \vee \text{charlas}[j].\text{fin} \leq \text{charlas}[i].\text{inicio}, \\ \text{false} & \text{en otro caso.} \end{cases}$$

Se debe entregar la solución antes del viernes 23 de octubre a las 17:00 vía EVA. El archivo debe estar en formato PDF y debe llamarse G-2-2.pdf, donde *G* es el número del grupo. Además se debe entregar una copia impresa en la clase de monitoreo de la semana que comienza el 26 de octubre.