

# Entrenamiento CP-UPV: Plan II

Competitive Programming UPV  
Generación Espontánea  
Universidad Politécnica de Valencia

`competitiveprogrammingupv@gmail.com`

*Revisado: 9 de febrero de 2022*

## 1. Duración del plan

- 1 semana
- Del sábado 26 de febrero al viernes 4 de marzo

## 2. Páginas a leer

- Desde la página 33 hasta la página 42.

Esta semana vamos a iniciar el estudio del segundo capítulo del libro “Competitive Programming 3”, “**Data Structures and Libraries**”. En este capítulo se explican las estructuras de datos y librerías que deberíamos dominar para poder obtener buenos resultados en competiciones de programación.

Específicamente, en esta semana estudiaremos estructuras de datos lineales para las que podemos encontrar implementaciones en librerías de varios lenguajes de programación.

En concreto, en el libro se hace énfasis en las estructuras de datos disponibles en C++, así que esta es una buena oportunidad para que aquellos que no estéis familiarizados con este lenguaje podáis mejorar vuestros conocimientos del mismo.

Como apunte, uno de los enlaces de la página 34 (*Visualization*: [www.comp.nus.edu.sg/~stevanha/visualization](http://www.comp.nus.edu.sg/~stevanha/visualization)) ha dejado de funcionar desde la publicación del libro, así que os indicamos el enlace a otra página web dedicada a la visualización de algoritmos y estructuras de datos: <https://visualgo.net/es>.

### 3. Ejercicios propuestos

A continuación, indicamos una serie de problemas que podéis intentar para practicar los conceptos explicados en esta primera parte del capítulo 2. Dado que en libro se explican en concreto las librerías de C++ que implementan las estructuras de datos presentadas, os recomendamos que, aunque no lo uséis normalmente, intentéis resolver los problemas propuestos usando este lenguaje.

- **Nivel 0**

⇒ [UVa 10038 - Jolly Jumpers] [https://onlinejudge.org/index.php?option=com\\_onlinejudge&Itemid=8&category=24&page=show\\_problem&problem=979](https://onlinejudge.org/index.php?option=com_onlinejudge&Itemid=8&category=24&page=show_problem&problem=979)

- **Nivel 1**

⇒ [UVa 146 - ID Codes] [https://onlinejudge.org/index.php?option=com\\_onlinejudge&Itemid=8&category=24&page=show\\_problem&problem=82](https://onlinejudge.org/index.php?option=com_onlinejudge&Itemid=8&category=24&page=show_problem&problem=82)

⇒ [UVa 514 - Rails] [https://onlinejudge.org/index.php?option=com\\_onlinejudge&Itemid=8&category=24&page=show\\_problem&problem=455](https://onlinejudge.org/index.php?option=com_onlinejudge&Itemid=8&category=24&page=show_problem&problem=455)

- **Nivel 2**

⇒ [UVa 11034 - Ferry Loading IV] [https://onlinejudge.org/index.php?option=com\\_onlinejudge&Itemid=8&category=24&page=show\\_problem&problem=1975](https://onlinejudge.org/index.php?option=com_onlinejudge&Itemid=8&category=24&page=show_problem&problem=1975)

⇒ [UVa 11933 - Splitting Numbers] [https://onlinejudge.org/index.php?option=com\\_onlinejudge&Itemid=8&category=24&page=show\\_problem&problem=3084](https://onlinejudge.org/index.php?option=com_onlinejudge&Itemid=8&category=24&page=show_problem&problem=3084)

- **Nivel 3**

⇒ [UVa 10920 - Spiral Tap] [https://onlinejudge.org/index.php?option=com\\_onlinejudge&Itemid=8&category=24&page=show\\_problem&problem=1861](https://onlinejudge.org/index.php?option=com_onlinejudge&Itemid=8&category=24&page=show_problem&problem=1861)

## 4. Soluciones

Hemos colgado las soluciones en nuestro GitHub: [https://github.com/cp-upv/cp3\\_solutions/tree/main/chapter2/semana2](https://github.com/cp-upv/cp3_solutions/tree/main/chapter2/semana2)

**¡Importante!** Recordad que para poder ver las soluciones deberéis ser miembros de nuestra organización en GitHub. Os podéis unir a través de este enlace: <https://cp-upv.herokuapp.com/>

Nuestras soluciones están realizadas mayoritariamente en C++ y algunas en Java o Python, pero no hay ningún problema en que utilicéis cualquiera de los lenguajes soportados por la UVa.

