

Entrenamiento CP-UPV: Plan IV

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 12 de marzo al viernes 18 de marzo

2. Páginas a leer

- Desde la página 49 hasta la página 54.

Esta semana vamos a continuar 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.

En concreto, en esta semana empezaremos a estudiar estructuras de datos que no tienen una implementación predeterminada y que, por tanto, deberemos implementar nosotros mismos.

La primera de estas estructuras que estudiaremos esta semana son los *Grafos*. Esta es posiblemente la estructura más importante de esta parte del libro, ya que prácticamente en cualquier competición habrá por lo menos un problema que requiera su uso.

En este capítulo solo aprenderemos como implementarla, pero más adelante en el libro encontraremos un capítulo entero (capítulo 4) dedicado a los diferentes algoritmos relacionados con Grafos.

La segunda estructura que estudiaremos son los *Union-Find Disjoint Sets*. Esta es una estructura que sirve para representar una colección de conjuntos disjuntos, de forma que las operaciones para determinar si un elemento pertenece a un conjunto en concreto, si dos elementos pertenecen al mismo conjunto (**find**) y la operación de unir dos subconjuntos en un único conjunto (**union**) están implementadas eficientemente.

Recordad que tenéis disponible la página web <https://visualgo.net/es>, en la que podéis visualizar cómo funcionan estas estructuras de datos por si no os queda claro su funcionamiento tras leer el libro.

3. Ejercicios propuestos

A continuación, indicamos una serie de problemas que podéis intentar para practicar los conceptos explicados en esta 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 2**

⇒ [UVa 599 - The Forrest for the Trees] https://onlinejudge.org/index.php?option=com_onlinejudge&Itemid=8&category=24&page=show_problem&problem=540

⇒ [UVa 11991 - Easy Problem from Rujia Liu?] https://onlinejudge.org/index.php?option=com_onlinejudge&Itemid=8&category=24&page=show_problem&problem=3142

⇒ [UVa 793 - Network Connections] https://onlinejudge.org/index.php?option=com_onlinejudge&Itemid=8&category=24&page=show_problem&problem=734

⇒ [UVa 11503 - Virtual Friends] https://onlinejudge.org/index.php?option=com_onlinejudge&Itemid=8&category=24&page=show_problem&problem=2498

⇒ [UVa 10608 - Friends] https://onlinejudge.org/index.php?option=com_onlinejudge&Itemid=8&category=24&page=show_problem&problem=1549

4. Soluciones

Hemos colgado las soluciones en nuestro GitHub: https://github.com/cp-upv/cp3_solutions/tree/main/chapter2/semana5

¡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.

