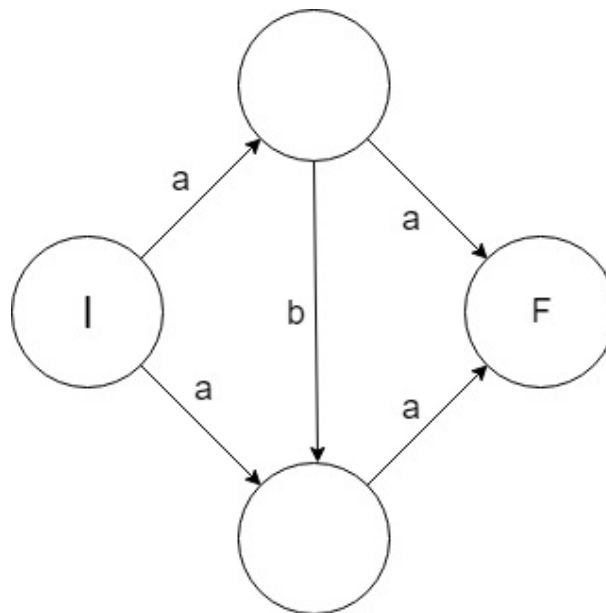


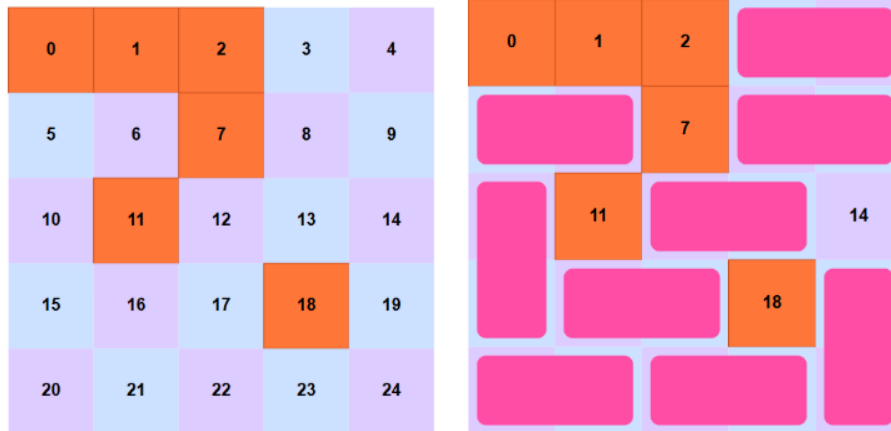
Ayudantía-Taller Flujo máximo y ordenación en $O(n)$

1. Flujo máximo

- a) Explique cómo se relaciona el flujo máximo entre dos nodos P y Q de un grafo con los cortes del grafo que dejan a P y Q separados.
- b) Explique por qué la complejidad del algoritmo de Ford-Fulkerson en el peor caso puede depender del peso de las aristas usando el siguiente grafo:



- c) Explique por qué los algoritmos de flujo máximo sirven para resolver problemas de emparejamiento.
- d) Tenemos un tablero como el de la figura y queremos saber cual es el máximo de dominós que podemos poner en el tablero para rellenarlo. Explique cómo resolver este problema como un problema de matching usando flujo máximo.



También explique qué significa el paso de flujo al traducirlo al tablero y qué significa pasar por una arista backward.

2. Ordenación en $O(n)$

2.1. Counting sort

- ¿Es counting sort estable? Justifique
- ¿Por qué counting sort puede ordenar datos más rápido que $O(n \cdot \log(n))$? ¿Qué contras tiene el algoritmo?
- ¿Qué pasa si se trata de usar counting sort con una lista de números reales?

2.2. Radix sort

- Explique qué pasa si no se usa un algoritmo de ordenación estable como sub rutina de radix sort en su versión LSD.
- Explique por qué radix sort MSD es un buen algoritmo de ordenación para strings.
- Explique por qué es conveniente cambiar de algoritmo de ordenación cuando hay pocos datos a ordenar en la versión MSD.
- (Actividad) Ordene los siguientes números usando radix sort desde el dígito menos significativo al más significativo.
 - 3646
 - 3276
 - 8445
 - 4687

- 2346
- 2386
- 2323
- 2335
- 2846
- 2723
- 2723
- 2623
- 3846