Guía Preexamen

Profesores: Iván Sipiran
Nelson Baloian
Patricio Poblete
Auxiliares: Alonso Almendras, Albani Olivieri
Vicente Olivares, Ricardo Valdivia
Sebastián Acuña, Martín Paredes

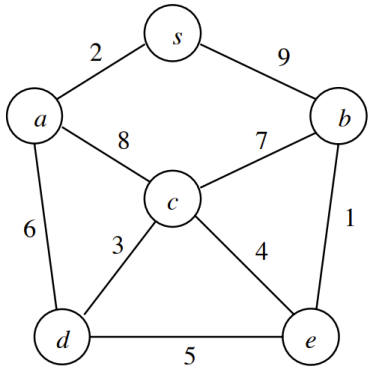
P1. Ex1 P1 2020-1

Suponga que a un arreglo que contiene elementos en orden estrictamente ascendente se le hace un desplazamiento cíclico, esto es, se empujan todos los elementos cero o más posiciones hacia la derecha, y los que salen por el extremo derecho reingresan por el extremo izquierdo. Por ejemplo, el siguiente sería el resultado después de que una lista ordenada se desplazó cíclicamente tres posiciones hacia la derecha:

Escriba una función `findmin` que reciba como parámetro un arreglo de este tipo, y retorne la posición del mínimo. En el caso del arreglo del ejemplo, debería retornar 3. Su algoritmo debería funcionar en tiempo estrictamente menor que $\Theta(n)$. Diga cuánto tiempo demora su algoritmo (en orden de magnitud) y justifique su respuesta.

P2. Ex1 P3 2020-1

Considere el siguiente grafo:



1. Aplique el algoritmo de Kruskal para encontrar un árbol cobertor de costo mínimo. Dibuje solo el árbol resultante y numere los arcos según el orden en que fueron siendo agregados.

2. Repita lo anterior usando el algoritmo de Prim.

P3. Ex1 P4 2020-1

Se desea mezclar dos listas enlazadas, cada una de las cuales viene ordenada en orden *ascendente* para formar una sola lista enlazada, ordenada en orden *descendente*. Escriba una función `merge` que reciba como parámetro dos listas a y b, y retorne la lista resultante. La función debe formar la lista de salida utilizando los mismos nodos de las listas de entrada, sin crear nuevos nodos. Por lo tanto, al terminar, las listas de entrada deben quedar vacías. La función debe correr en tiempo proporcional al largo de la lista de salida.

Ejemplo:

P4. Ex2 P1 2020-1

Resuelva la siguiente ecuación de recurrencia:

$$a_n = a_{n-1} + 12a_{n-2}$$
$$a_0 = 0$$
$$a_1 = 7$$

P5. Ex2 P2 2020-1

Suponga que se tiene un grafo G no dirigido. Modifique las funciones `DFS` y `startDFS` para que encuentren la componente conexa que tiene más nodos. La función modificada `startDFS` debe retornar el número de nodos de dicha componente.

P6. Ex2 P3 2020-1

Se tiene una lista enlazada que se desea particionar al estilo Quicksort. Para esto, se debe tomar el primer elemento como pivote, y luego recorrer el resto de la lista construyendo dos listas enlazadas: una con los elementos menores que el pivote y la otra con los elementos mayores o iguales que el pivote. La función debe formar la listas de salida utilizando los mismos nodos de la listas de entrada, sin crear nuevos nodos. Por lo tanto, al terminar, la lista de entrada debe quedar vacía. Más concretamente, usted debe escribir una función `particiona` que reciba como parámetro una lista y retorne una tupla que tenga (lista de los menores, pivote, lista de los mayores), donde "pivote" es una lista de largo 1 que contiene solo al pivote.

P7. Boyer-Moore (BM)

BMH (Boyer-Moore-Hospool) compara el patrón buscado con el texto, leyendo este último de izquierda a derecha, pero comparando el patrón de derecha a izquierda.

Considere el siguiente texto:

```
TRES TRISTES TIGRES TRIGO TRIGABAN EN UN TRIGAL
```

Busque la secuencia TRIG, mediante el algoritmo Boyer-Moore-Hospool (BMH), y cada vez que la encuentre, aumente el contador en uno.

¿Cuántas veces aparece la secuencia en el texto? ¿Cuál es el costo de buscar secuencias con este algoritmo? ¿Existe alguna manera un poco más eficiente para realizar lo mismo? Si existe, repita el mismo procedimiento anterior, pero usando el nuevo algoritmo.