



Facultad de Ciencia

Departamento de Matemática y Ciencia de la Computación

Paradigmas de programación

Semestre 2020-01

Ejercicios de ayudantía

Nicolás Honorato Droguett
Matías Hurtado Carrasco

Pablo Perez Lantero
12 de mayo de 2020

Índice

1	2020-05-11	2
1.1	Ayudantía Nicolás	2
	Ejercicios	2
1.2	Ayudantía Matías	2
	Ejercicios	2

1 2020-05-11

1.1 Ayudantía Nicolás

Ejercicios

Pregunta 1 Programe en `Python` la función `bloques(g, key)` donde `g` es un generador de elementos, y `key` es una función de un argumento que recibe elementos de `g` y devuelve valores comparables entre sí usando los operadores habituales de comparación. La función genera los bloques de elementos consecutivos generados por `g` que están ordenados ascendentemente de acuerdo al valor de `key`. Cada vez que se rompe el orden se genera un nuevo bloque. Cada bloque es generado como una lista. Por ejemplo, si `g` genera los elementos `1,5,6,4,8,9,2,3,7` en este orden y `key` es la función identidad, entonces se generarán los bloques `[1,5,6]`, `[4,8,9]`, `[2,3,7]` en este mismo orden. Tenga presente en la implementación que `g` podría generar infinitos elementos, o una cantidad finita pero extremadamente grande.

Pregunta 2 La función `merge(A,B)` recibe dos listas `A` y `B` de números, cada una ordenada ascendentemente, y devuelve una tercera lista con la mezcla ordenada de `A` y `B`. Usando la función `merge`, realice dos implementaciones distintas de la función `mega_merge(*L)` que recibe varias listas de números como parámetros, cada lista ordenada, y devuelve la mezcla de todas las listas. La primera implementación debe usar la función `reduce`, mientras que la segunda implementación no puede usarla.

1.2 Ayudantía Matías

Ejercicios

Pregunta 1 **Pregunta 2**