

Reporte Tarea 2

Fermin A. Ahumada Garcia

September 7, 2022

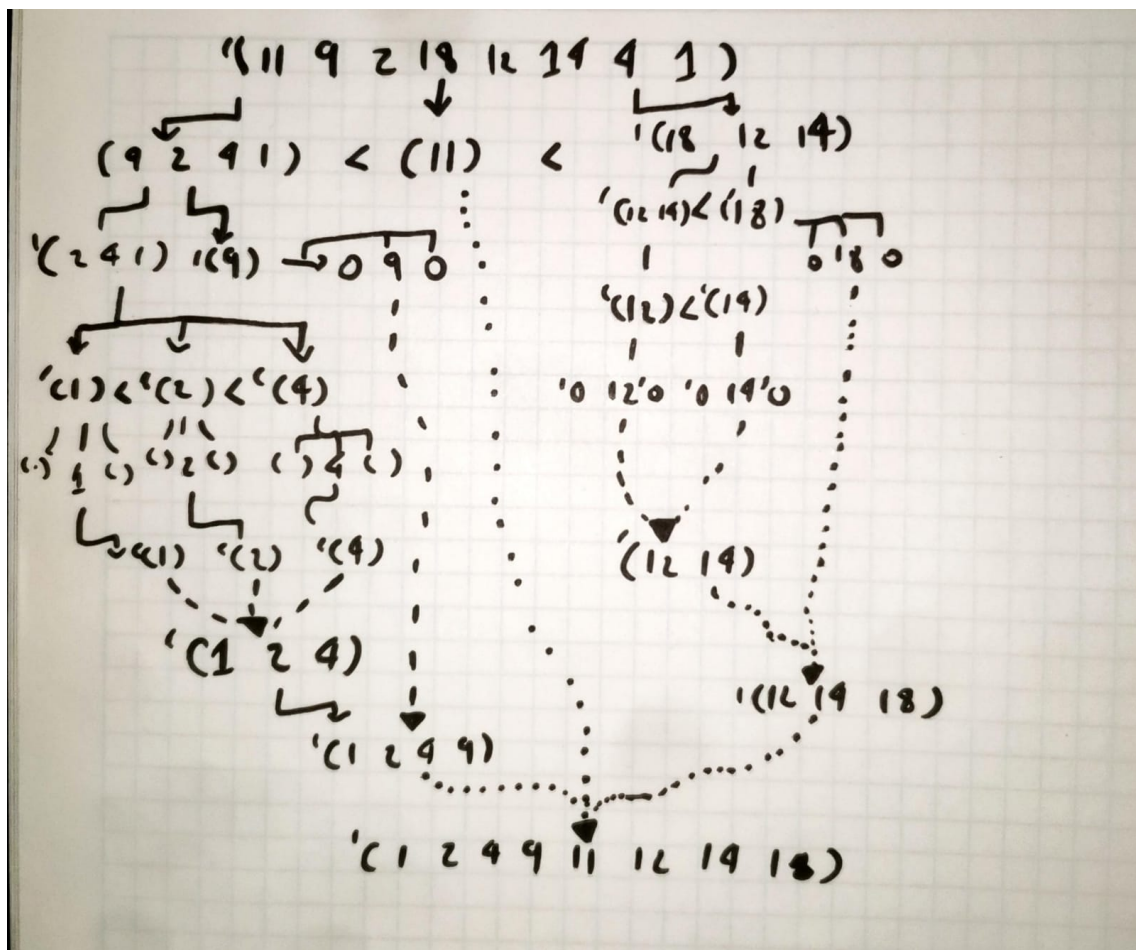
Problema 5

La llamada `(bundle '("a" "b" "c") 0)` es un buen uso de `bundle`? ¿qué produce? ¿por qué?

Este ejemplo en particular iba generar un bucle infinito, por que al llamar a *drop* y *take*, al ser $n = 0$, hace que no recorra los elementos de esta, y nunca se vaciará.

Problema 9

Dibuja un diagrama como el de la figura anterior pero para la lista `'(11 9 2 18 12 14 4 1)`.



Problema 11

Si la entrada a quicksort contiene varias repeticiones de un número, va a regresar una lista estrictamente más corta que la entrada. Responde el por qué y arregla el problema. La entrada a quicksort al tener una gran cantidad de números repetidos, esta va a regresar una lista más corta que la entrada. Al entrar en los métodos de *largers* y *smallers* solo agregan los elementos que cumplan al ser mayor o menor que el pivote. En algún punto de la recursión, el pivote repetido estará en la lista, lo cual el algoritmo lo va a ignorar por no saber que hacer con el. Por eso se creo una función llamada *eq*, que ingresa estos valores repetidos.

```
1 (define (eq ls n)
2   (if (null? ls)
3       null
4       (cond
5         [(eq? (first ls) n) (cons n (eq (rest ls) n))]
6         [else (eq (rest ls) n)])))
```

Esta función filtrará a los resultados que son iguales de ls al pivot, y los agregará a una lista nueva, modificando *quicksort* para juntar 3 listas