

Alumnos:
 Garcia Garcia Raul Eduardo
 Rivera Mercado Sergio
 Rivera Sotomayor Luis Rafael

1.-

1.1

(Subst '{with{x 3}

{with{y 2}

{with {z {+ 1 w}

{ + { x { + y z } } } w 0}

+ { x { + y z } } } w 0
z = w+1
y=2
x=3

1.2

Heap sort ahorraría operaciones ya que la inserción y análisis de datos estarían manejados por una sola estructura.

1.3

Tomando en cuenta el código del punto 1, insertamos un with en nuestra nueva estructura de datos, al tratarse de un heap, podemos garantizar que en casos promedios el orden de subst es igual al de la estructura previa, pero como en el peor de los casos, como nuestra estructura se trata de un heap y toda operación está implementada por el, sería a los mas $O(n \log n)$

2.-

No es verdad, dando un contraejemplo tenemos:

{with { x 4}

{with { x 8}

{with { f { fun { y } { + x y } } }

{ with { x 5 }

{ f 10 } } }

f 10
x 5
f { fun { y } { + x y } }
x 8
x 4

Si hacemos alcance dinámico al momento de evaluar la función tendremos num 18, pues debajo de esta tenemos x 8, pero x 4 es el valor más viejo de x.

3.-

3.1

Alumnos:

Garcia Garcia Raul Eduardo

Rivera Mercado Sergio

Rivera Sotomayor Luis Rafael

$\lambda x(y (x y) (2\ 3))$

$\lambda y (5 x)$

$\lambda x (10 z (y (5)) ((y\ x)\ z))$

->

$\lambda x(5 (1 (5,2)) (2\ 3))$

$\lambda y (5\ 1)$

$\lambda x (10\ 1 (2 (5)) ((1\ 2)\ 3))$