

6. (a)2 (b)4 (c)3

7. (a)8

(b)

(1)	access link	(0)		
	x	1		
(2)	access link	(1)	⟨ (.) ⟩	code for f
	f	•		...
(3)	access link	(2)		
	q	2	⟨ (.) ⟩	code for h
(4)	access link	(3)		
	h	•		
(5)	access link	(4)		
	w	8		
(6)	h(3)	access link	(5)	
	z	3		
(7)	f(3)	access link	(6)	
	y	3		

8.

(a)

Activation Records

(1)	access link	(0)
	f	•
	h	•
(2)	h(2)	access link
	x	2
(3)	f(1)	access link
	x	1

(b) B

(c) JavaScript 采用 statically scope , 由于 :

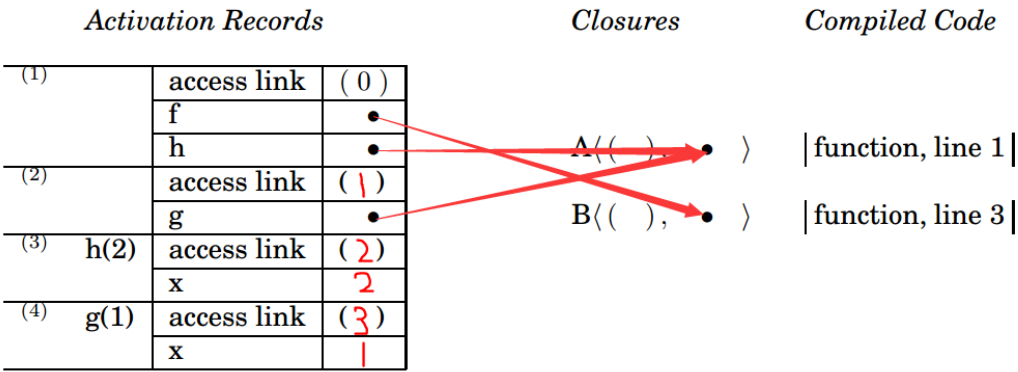
Un-named recursive function: In the first declaration, variable `f1` is initialized to an anonymous function which has recursive call to `f1`. When this line of code is executed, a closure for the function is created. The environment pointer of this closure points to the activation record of the scope where `f1` is declared.

所以：

line 2 后面调用 `h (2)`, `h ()` 里的 `f` lookup 到的是原来的 `f1 ()`, 故为 2 ；

而 line3 后调用 `h (2)`, `h ()` 里的 `f` lookup 到的是新的 `f3 ()`, 故为 20。

(d)



(e)

由于 `f1 ()` 中递归的是 `g ()`, 而不是 “ `f` ” ,

所以 `h (2)` 调用的是 `g (2)`,

Return `2*g(1)->2*1->2` , 所以返回 2.