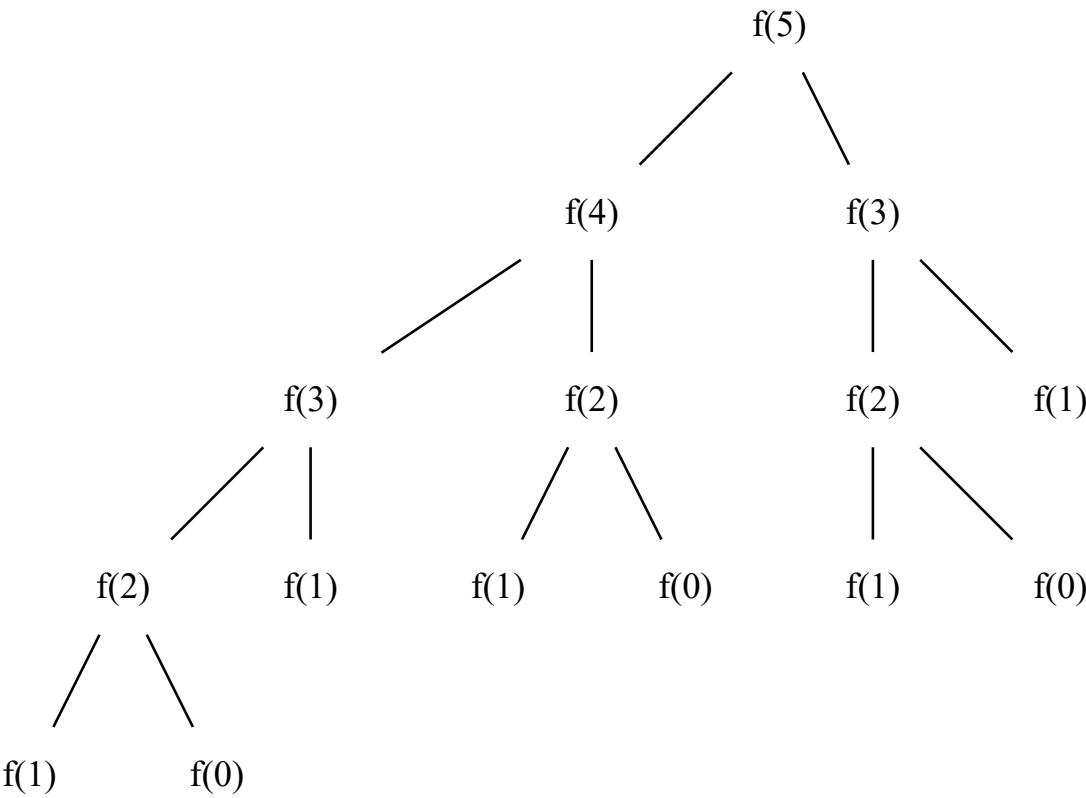


编译原理作业 13

23.12.05

7.2.3

(1)



(2)

返回值	n	s	t
?	5	?	?
?	4	?	?
?	3	?	?
?	2	?	?
1	1	?	?

为了节省空间, 我将一个过程的活动记录横向书写, 将其竖向摆放即为栈的形状.

问号表示未初始化, 可能是之前遗留的任意的值.

(3)

返回值	n	s	t
?	5	5	?

返回值	n	s	t
? (5)	3	2	? (2)
1	1	? (1)	? (1)

有些内存单元虽未初始化, 但根据之前的调用记录, 可以推断出遗留的值, 标注在了括号中.

7.2.6

返回值是 $7 + 7 + 7$, 即 21.

7.3.1

活动记录栈从栈底到栈顶, 分别是 `main` `fib0(4)` `fib1(4)` `fib2(4)` `fib1(3)` `fib0(2)` `fib1(2)` `fib0(1)` 的栈帧.

标号	0	1	2	3	4	5	6	7	栈顶
函数	main	fib0(4)	fib1(4)	fib2(4)	fib1(3)	fib0(2)	fib1(2)	fib0(1)	
访问链	?	0	1	2	1	0	1	0	

7.5.1

- (1) B 的引用计数减 1, B 被回收, 然后 D, E 的引用计数减 1, D 被回收, 然后 G 的引用计数减 1.
- (2) A 的引用计数减 1, A 被回收, 然后 B, C 的引用计数减 1, B 被回收, 然后 D, E 的引用计数减 1, D 被回收, 然后 G 的引用计数减 1.
- (3) F 的引用计数减 1, F 被回收, 然后 H 的引用计数减 1.

7.6.1

- (1) ACEFGHI 被标记, BD 被回收.
- (2) 所有对象都被标记, 没有对象被回收.
- (3) ABCEHI 被标记, DFG 被回收.
- (4) ADEFGHI 被标记, BC 被回收.