```
(define-struct spider (legs space))
;make-spider: number number --> spider
;spider-legs: spider --> number
;spider-space: spider --> number
;spider?: object --> boolean
(define spider1 (make-spider 7 1))

(define-struct elephant (space))
;make-elephant: number --> elephant
;elephant-space: elephant --> number
;elephant?: object --> boolean
(define elephant1 (make-elephant 2500))

(define-struct monkey (IQ space))
;make-monkey: number number --> monkey
;monkey-IQ: monkey --> number
;monkey-space: monkey --> number
;monkey?: object --> boolean
(define monkey1 (make-monkey 100 125))

(define (fits? animal cage)
  (cond
    [(spider? animal) (< (spider-space animal)
cage)]
    [(elephant? animal) (< (elephant-space
animal) cage)]
    [(monkey? animal) (< (monkey-space animal)
cage)]
    [else false]))

(check-expect (fits? spider1 3) true)
```

```
(check-expect (fits? elephant1 2000) false)
(check-expect (fits? monkey1 200) true)
(check-expect (fits? "Zebra" 500) false)
```