lib/basic/rsarray.ath

```
# A rudimentary implementation of resizable arrays.
# The argument "increment" given to the constructor
# make-rs-array denotes the minimum chunk of memory
# that will be allocated on expansion.

(define (make-rs-array size init-element increment)
  [(cell (make-vector size init-element)) (cell size) increment])

(define rs-array-vector-cell first)

(define rs-array-size-cell second)

(define (rs-array-size A) (ref (rs-array-size-cell A)))

(define rs-array-increment third)

(define (rs-array-sub A i)
  (vector-sub (ref (rs-array-vector-cell A)) (minus i 1)))

(define (rs-array-set A i val)
  (let ((size (rs-array-size A)))
    (check ((greater? i size)
            (let ((increment (rs-array-increment A))
                  (diff (minus i size))
                  (minumum-needed (div diff increment))
                  (increments-needed (check ((equal? (mod diff increment) 0) minumum-needed)
                                            (else (plus 1 minumum-needed))))
                  (new-size (plus size (times increments-needed increment)))
                  (V' (make-vector new-size ()))
                  (_ (vector-copy (ref (rs-array-vector-cell A))
                                  V'))
                  (_ (set! (rs-array-size-cell A) new-size))
                  (_ (vector-set! V' (minus i 1) val)))
              (set! (rs-array-vector-cell A) V')))
           (else (vector-set! (ref (rs-array-vector-cell A)) (minus i 1) val)))))
```