**Exercise 9.2**

1. **Show that RANDOMIZED-SELECT never makes a recursive call to a 0-length array.**

**Ans)**

The random select, function is as follows: RANDOM-SELECT(A, p , r , i):

if p == r:

return A[p]

else

q = RANDOM-PARTITION(A,p,r)

k = q – p + 1

if k == i:

return A[q]

else if i < k:

return RANDOM-SELECT(A, p, q-1, i)

else

return RANDOM-SELECT(A,q+1,r, i – k)

Here, recursive calls will be made to 0 length array if p = r = 0

1. p = q-1 = 0

* q-p +1 = 0
* k = 0
* However, this condition is only reached if i < k but we know i > 0. Therefore this condition will never be reached.

1. i>k

* i > q – p +1
* i > r – p
* Since array is size 0, r = p = 0 will always hold true and this condition will never be met because no partitioning is happening at this step.

1. **Write an iterative version of RANDOM-SELECT.**

**Ans)**

RANDOM-SELECT(A,p,r,i):

if p == r:

return A[p]

while p<r:

q = RANDOM-PARTITION(A,p,r)

k = q – p + 1

if k == i:

return A[q]

else if k < i:

r = q-1

else:

p = q + 1

i = i - k

1. **Suppose that RANDOMIZED-SELECT is used to select**

**Ans)**

<9,8,7,6,5,4,3,2,1,0>

1. **Argue that the expected runtime of RANDOMIZED-SELECT does not depend on the order of the elements in its input array A[p : r]. That is, the expected running time is the same for nay permutation of the input array A[p:r].**

**Ans)**