Bonus Questions (7.1) F

Initially a procedure to find the Euclian distance between a point an orisin.

procedure Edist(A)
  return $\sqrt{A_1^2 + A_2^2}$ // the two points inside A
End procedure.

Now using a simple bucketsort to do the job. (simple pseudocode copied from slide)

Procedure Bucketsort (A)
Let B[0.... n-1] be a new array
n = A.length.
~~Fo~~ FOR I = 0 to n-1
  B[I] ← 0
END FOR.
FOR I = 1 to N
  Temp = Edist(A[I])
  B[n x Temp] = Temp
END FOR
FOR i = 0 to n-1
  sort list B[I] using insertion sort
END FOR
  concatenate the lists from B[0] ... B[n-1].
End Procedure.

The pseudocode for computation of Euclidian distance in 2 points is.

Procedure Dist (A, B)
return $\sqrt{(A_1-B_1)^2+(A_2-B_2)^2}$
END procedure.

Problem 7.2

(b) we can use a simple Radix sort to do the job. we know ~~each~~ countsort takes ~~to~~ O(n) to compare integers so to find the nom of digits is $\log_n n^3$

(c) To do this Firstly we need to go through

c) RadixSort (A)
A ← Convert to base$_n$ (A)
FOR I = 1 to 3
Countsort (A, I)
END FOR
END PROCEDURE.

Fiist just go through the entire list ~~convo~~ and convert each number to base n. Then use radix sort on them and we know there will be ~~3 passes~~ may 3 digits as $\log_n n^3 = 3$. ~~So~~ ~~as~~ There complexity is ~~O~~ O(3n) = O(n) //.