University of Houston

Programming Assignment 2

COSC 3320
Algorithms and Data Structures

Gopal Pandurangan

Solution

1 Pseudocode and Explanation

Notice that this problem is effectively just the Select algorithm on the *distances from the origin*. Letting $\text{dist}_0^2(x, y) = x^2 + y^2$, our algorithm is simply

alg:selection

Algorithm 1 Select

def $\textsc{Select}(\texttt{points}, k)$:
 Input ▷ An array `points` of n ordered pairs and the desired order statistic, k
 Output ▷ The value of the k^{th} order statistic
 if $|\texttt{points}| \leq 1$:
  return $\texttt{points}[0]$
 else:
  Partition `points` into $\lfloor n/5 \rfloor$ groups of 5 elements each and a leftover group of up to 4 elements.
  Find the median of each group using MergeSort with comparison function dist_0^2
  $\texttt{medians} \leftarrow$ the set of these $\lfloor n/5 \rfloor$ medians
  $\texttt{pivot} \leftarrow \textsc{Select}(\texttt{medians}, \lfloor n/10 \rfloor)$
  $\texttt{left} \leftarrow \left\{p \in \texttt{points} \mid \text{dist}_0^2(p) < \text{dist}_0^2(\texttt{pivot})\right\}$
  $\texttt{right} \leftarrow \left\{p \in \texttt{points} \mid \text{dist}_0^2(p) > \text{dist}_0^2(\texttt{pivot})\right\}$
  if $|\texttt{left}| = k - 1$:
   return p
  else if $|\texttt{left}| > k - 1$:
   return $\textsc{Select}(\texttt{left}, k)$
  else:
   return $\textsc{Select}(\texttt{right}, k - |\texttt{left}| - 1)$

Then we can simply return the k closest points to the origin by comparing with $\textsc{Select}(\texttt{points}, k)$. However, we must be careful about duplicates: say there are ℓ points less than the k^{th} closest point and t points equidistant. Then, by definition, we must have $\ell + t \geq k$. If $t = 1$ then $\ell = k - 1$ (of which all distances being distinct is a special case). Either way, we must choose all ℓ points and then any $k - \ell$ points equidistant to our k^{th} closest point.

$$\underbrace{\underbrace{\{p_1, p_2, \ldots, p_\ell\}}_{\ell \text{ points}} \underbrace{\{p_{\ell+1}, p_{\ell+2}, \ldots, p_k}_{k-\ell \text{ points}}}_{\ell+(k-\ell)=k \text{ points}}, \ldots, p_{\ell+t}\}$$

alg:kclosest

Algorithm 2 k-Closests

def $k\text{-}\textsc{Closests}(\texttt{points}, k)$:
 Input ▷ An array `points` of n ordered pairs and an integer k
 Output ▷ The k closest points to the origin
 if $|\texttt{points}| \leq k$:
  return `points`
 else:
  $\texttt{cutoff} \leftarrow \textsc{Select}(\texttt{points}, k)$
  $\texttt{left} \leftarrow \left\{p \in \texttt{points} \mid \text{dist}_0^2(p) < \text{dist}_0^2(\texttt{cutoff})\right\}$
  $\texttt{equal} \leftarrow \left\{p \in \texttt{points} \mid \text{dist}_0^2(p) = \text{dist}_0^2(\texttt{cutoff})\right\}$
  $\ell \leftarrow |\texttt{left}|$
  append $k - \ell$ points of `equal` to `left`
  return `left`

2 Analysis

Correctness follows from correctness of the Select from the textbook. The runtime of Select follows from the same analysis. Specifically, our runtime is $\Theta(n)$.