Selection in worst-case linear time

ESO207

Indian Institute of Technology, Kanpur

Overview

- Recall: Selection problem SELECT($A, 1, n, i$)–find the ith smallest element in $A[1 \ldots n]$.
- We will now see a selection algorithm that runs in time $O(n)$.
- Like the previous RAND-SELECT, this algorithm uses the *Partition* algorithm and recursively partitions the array.
- But the algorithm is deterministic–pivot selection is deterministic (no randomness).

Algorithm steps

Algorithm $\textsc{Select}(A, 1, n, i)$

1. Divide the n elements of the input array into $\lceil n/5 \rceil$ groups of 5 elements.
   - All groups have 5 elements, except perhaps the last that has $n \mod 5$ elements.
2. Find the median of each of the $\lceil n/5 \rceil$ groups.
   - Do an insertion-sort on the elements of each group—each group has at most 5 elements.
3. Recursively call $\textsc{Select}$ to find the median x of the $\lceil n/5 \rceil$ medians found in Step 2.
   - If there are an even number of medians (groups), choose the lower median.

Selection Algorithm (contd).

4. Partition the input array around the median-of-medians x by choosing x as the pivot.
5. Let k be one more than the number of elements in the left partition, so that x is the kth smallest element and there are $n - k$ elements in the right partition.
6. If $i = k$ then return x. Otherwise, if $i < k$, call SELECT recursively on the left partition, and if $i > k$, call SELECT recursively to find the $i - k$th smallest element on the right partition.

Analysis: Median of medians algorithm

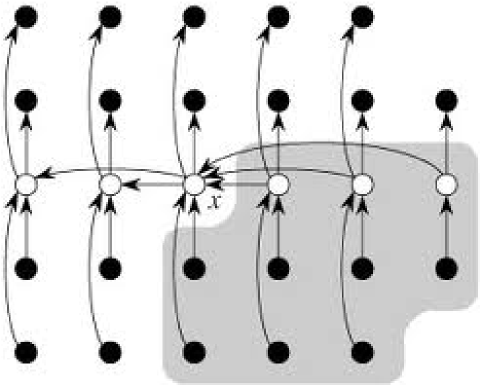

Figure: Depiction of Median-of Median algorithm for Selection. Elements are circles, medians of groups are white circles. Arrows go from larger to smaller elements. x is the median of the medians. Elements known to be greater than x are shaded.

Analysis

- Assume that the numbers are distinct. We first determine a lower bound on the number of elements that are greater than the partitioning element x.
- At least $1/2$ the medians of the groups are greater than or equal to the median-of-medians x. So at least half of the $\lceil n/5 \rceil$ groups contribute at least 3 elements that are greater than x, except for
  - one group that has fewer than 5 elements if 5 does not divide n exactly, and,
  - the one group that contains x itself contributes 2 elements
- Thus, number of elements greater than x is at least

$$3\left(\left\lceil \frac{1}{2}\left\lceil \frac{n}{5} \right\rceil \right\rceil - 2\right) + 2 \geq \frac{3n}{10} - 4$$

Analysis: Median-of-medians Selection

- In a similar manner, it can be shown that there are $3n/10 - 4$ elements in A that are less than x.
- Using x as the pivot, SELECT calls recursively on at most $7n/10 + 4$ elements.

Analysis: Recurrence Equation

Let us now derive the recurrence equation for the worst-case running time $T(n)$ of SELECT.

1. Finding the median of each group of 5 elements, can be done in time $O(n)$.
2. Partitioning the input array around the median-of-medians x is also done in time $O(n)$.
3. Using SELECT recursively to find the median x of the $\lceil n/5 \rceil$ medians is done in time $T(\lceil n/5 \rceil)$.
4. Finally, the recursive SELECTprocedure takes time at most $T(\lceil 7n/10 \rceil + 4)$, assuming that T is monotonically increasing.

Recurrence Equation

- The recurrence equation for the time complexity $T(n)$ is

$$T(n) \leq \begin{cases} O(1) & \text{if } n \leq 100 \\ T(\lceil n/5 \rceil) + T(\lceil 7n/10 \rceil + 4) + O(n) & \text{otherwise.} \end{cases}$$

- Here, 100 is a magic constant whose value will be derived in the analysis.

Solving the recurrence

- We will solve the recurrence

$$T(n) \le T(\lceil n/5 \rceil) + T(7n/10 + 4) + an \ .$$

  by substitution.
- Let $T(n) \le cn$ for some suitably large constant c. Assume that $T(n) \le cn$ for some suitably large c and all $n < 100$. This holds for c large enough.
- The $O(n)$ term is replaced by an upper bound an, for $n > 0$.

Solving the recurrence

- We will solve the recurrence

$$T(n) \leq T(\lceil n/5 \rceil) + T(7n/10 + 4) + an \ .$$

  by substitution.
- Let $T(n) \leq cn$ for some suitably large constant c. Assume that $T(n) \leq cn$ for some suitably large c and all $n < 100$. This holds for c large enough.
- The $O(n)$ term is replaced by an upper bound an, for $n > 0$.

Solving the recurrence

- We will solve the recurrence

$$T(n) \leq T(\lceil n/5 \rceil) + T(7n/10 + 4) + an \ .$$

  by substitution.
- Let $T(n) \leq cn$ for some suitably large constant c. Assume that $T(n) \leq cn$ for some suitably large c and all $n < 100$. This holds for c large enough.
- The $O(n)$ term is replaced by an upper bound an, for $n > 0$.

Solving the recurrence

- Substituting $T(n) \leq cn$ in the *RHS* of the equation

$$T(n) \leq T(\lceil n/5 \rceil) + T(\lceil 7n/10 \rceil + 4) + an \ .$$

- we get

$$\begin{aligned} T(n) &\leq c\lceil n/5 \rceil + c(7n/10 + 4) + an \\ &\leq cn/5 + c + 7cn/10 + 4c + an \\ &= 9cn/10 + 5c + an \\ &= cn + (-cn/10 + 5c + an) \end{aligned}$$

- which is at most cn if

$$-cn/10 + 5c + an \leq 0$$

Solving the recurrence

- Substituting $T(n) \leq cn$ in the *RHS* of the equation

$$T(n) \leq T(\lceil n/5 \rceil) + T(\lceil 7n/10 \rceil + 4) + an \quad .$$

- we get

$$\begin{aligned} T(n) &\leq c\lceil n/5 \rceil + c(7n/10 + 4) + an \\ &\leq cn/5 + c + 7cn/10 + 4c + an \\ &= 9cn/10 + 5c + an \\ &= cn + (-cn/10 + 5c + an) \end{aligned}$$

- which is at most cn if

$$-cn/10 + 5c + an \leq 0$$

Solving the recurrence

- Substituting $T(n) \leq cn$ in the *RHS* of the equation

$$T(n) \leq T(\lceil n/5 \rceil) + T(\lceil 7n/10 \rceil + 4) + an \ .$$

- we get

$$\begin{aligned} T(n) &\leq c\lceil n/5 \rceil + c(7n/10 + 4) + an \\ &\leq cn/5 + c + 7cn/10 + 4c + an \\ &= 9cn/10 + 5c + an \\ &= cn + (-cn/10 + 5c + an) \end{aligned}$$

- which is at most cn if

$$-cn/10 + 5c + an \leq 0$$

Solving the recurrence

- The equation

$$cn/10 - 5c - an \geq 0$$

is equivalent to

$$c \geq \frac{10an}{n-50}, \quad \text{for } n > 50$$

- Choose $n \geq 100$ so that $n/(n-50) \leq 2$.
- Now choosing $c = 20a$ will satisfy the inequality, for $n \geq 100$.
- The worst-case running time of SELECT is therefore $O(n)$.