Order Statistics

Joshua Shim

October 2024

Problem: Given n elements, select the i^{th} smallest element.
Subproblem 1 - Finding i in linear time using a randomised algorithm:
Idea: Divide and Conquer; by doing one half of the randomised QuickSort, operating on one side of the partition only.
Rand-Select(A, p, r, i): choose the i^{th} smallest element of $A[p..r]$.

1. if $p = r \& i = 1$ return $A[p]$;
2. choose a random pivot $p \leq q \leq r$;
3. reorder the array $A[p..r]$ such that $A[p..(q-1)] \leq A[q]$ and $A[(q+1)..r] > A[q]$;
4. $k \leftarrow q - p + 1$
5. if $k = i$ return $A[q]$;
6. if $k < i$ return Rand-Select$(A, p, q-1, i)$;
7. return Rand-Select$(A, q+1, r, i-k)$.

Analysis of Rand-Select(A, p, r, i)
Clearly the worst case run time is $\Theta(n^2)$. However, this is very unlikely to happen:

- Let us first assume that all the elements in A are distinct.
- Let us call a partition a *balanced partition* if the ratio between the number of elements in the smaller piece and the number of elements in the larger piece is not worse than 1 to 9.

- what is the probability that we get a balanced partition after choosing the pivot?

- This happens if we chose an element which is neither among the smallest $\frac{1}{10}$ nor amongst the largest $\frac{1}{10}$ of all elements.

- Thus, the probability to end up with a balanced partition is $1 - \frac{2}{10} = \frac{8}{10}$.

- The probability that you will need k partitions to end up with another balanced partition is $(\frac{2}{10})^{k-1} \cdot \frac{8}{10}$.

- The expected number of partitions between two balanced partitions is

$$\begin{aligned} E &= 1 \cdot \frac{8}{10} + 2 \cdot \frac{2}{10}\frac{8}{10} + 3 \cdot (\frac{2}{10})^2 \cdot \frac{8}{10} + \ldots \\ &= \frac{8}{10} \cdot \sum_{k=0}^{\infty}(k+1)(\frac{2}{10})^k = \frac{8S}{10} \end{aligned}$$

  where

$$S = 1 + 2 \cdot \frac{2}{10} + 3 \cdot (\frac{2}{10})^2 + 4 \cdot (\frac{2}{10})^3 + 5 \cdot (\frac{2}{10})^4 + \ldots$$

- $S = (\frac{10}{8})^2$

- Thus we obtain

$$E = \frac{8S}{10} = \frac{8}{10} \cdot (\frac{8}{10})^2 = \frac{10}{8} = \frac{5}{2} < 2.$$

- So, on average, there are only $\frac{5}{4}$ partitions between two balanced partitions (such as how we defined it).

- Consequently, the total expected run time satisfies

$$\begin{aligned} T(n) &< \frac{5n}{4} + \frac{5}{4}\frac{9n}{10} + \frac{5}{4}(\frac{9}{10})^2 n + \frac{5}{4}(\frac{9}{10})^3 n + \ldots \\ &= \frac{\frac{5}{4}n}{1 - \frac{9}{10}} \\ &= \frac{50}{4}n \\ &= 12.5n \end{aligned}$$

Subproblem 2 - Finding i in linear time using a deterministic algorithm:
Det-Select(n, i):

1. Split the numbers in groups of five;
2. Order each group by brute force in an increasing order.
3. Take every $\lfloor \frac{n}{5} \rfloor$ middle elements of each group.
4. Recursively apply Det-Select algorithm to find the median p of this collection.
5. Partition all elements using p as a pivot;
6. Let k be the number of elements in the subset of all elements $< p$.
7. if $i = k$ return p.
8. if $i < k$ recursively SELECT the i^{th} smallest element of the set of elements smaller than the pivot.
9. otherwise $(i > k)$ recursively SELECT the $(i - k)^{th}$ smallest element of the set of elements larger than the pivot.

What have we accomplished by such a choice of pivot?
At least $\lfloor (\frac{n}{10}) \rfloor$ group medians are $\leq p$; and at least that many are larger than the pivot.
Analysis of Det-Select(i, n):

- What is the runtime of our algorithm?

$$T(n) \leq T(\frac{n}{5}) + T(\frac{7n}{10} + Cn)$$

- Let us show that $T(n) < 11Cn$ for all n. Assume that this is true for all $k < n$ and let us prove it is true for n as well.

- Thus, assume $T(\frac{n}{5}) < 11C \cdot \frac{n}{5}$ and $T(\frac{7n}{10}) < 11C \cdot \frac{7n}{10}$; then

$$\begin{aligned} T(n) &\leq T(\frac{n}{5}) + T(\frac{7n}{10}) + Cn \\ &< 11C \cdot \frac{n}{5} + 11C \cdot \frac{7n}{10} + Cn \\ &= 109\frac{Cn}{10} \\ &< 11C \cdot n \end{aligned}$$

  which proves our statement that $T(n) < 11C \cdot n$.