CS207 Design and Analysis of Algorithms

Sajith Gopalan

Indian Institute of Technology Guwahati

sajith@iitg.ac.in

January 21, 2022

Divide and Conquer

Divide and Conquer

- solve a problem recursively
  - Divide the problem into a number of smaller instances of the same problem
  - For each of these subproblems, if its size is sufficiently large, then Conquer it recursively, else Conquer it directly
  - Combine the solutions to the subproblems into the solution for the original problem

Selection

- Given is an array A of n elements drawn from a linearly ordered set
- Given is $k \in \{1, \ldots, n\}$
- Find the k-th smallest element in A
- A trivial solution: sort A; pick the k-th element in the sorted sequence
- This takes $O(n \log n)$ time.
- There is a D&C algorithm that solves this in $O(n)$ time: Blum's algorithm

Blum's Algorithm

Algorithm 1 Blum(A,k) /* wlg, elements of A are distinct */

if ($|A| \le n_0$) sort A and **return** the k-th smallest element
Visualize A as a 2-D array with 5 rows and $\lceil n/5 \rceil$ columns
Sort each column in ascending order
Create an array A' of size $\lceil n/5 \rceil$ by picking the median element of each column
Let $m = \text{Blum}(A', \lceil n/10 \rceil)$ /* Recursive call */
/* m is the median of A', because $\lceil \lceil n/5 \rceil /2 \rceil = \lceil n/10 \rceil$ */
Partition A into A_L, $\{m\}$, A_U so that A_L's elements are $< m$ and A_U's elements are $> m$; $|A_L| + 1 + |A_U| = n$
if ($k < |A_L|$) **return** Blum(A_L, k)
else if ($k > |A_L| + 1$) **return** Blum($A_U, k - |A_L| - 1$)
else return m;

Selection

Array A visualized as a 2-D array with 5 rows and $\lceil n/5 \rceil$ columns:

04	32	40	24	56	41	61	27	12	46
16	28	48	17	21	08	37	39	58	55
36	49	33	23	06	25	11	31	14	05
10	47	54	29	18	52	44	62	30	13
35	70	34	65	45	67	57	03	02	

Selection

04	28	33	17	06	08	11	03	02	05
10	32	34	23	18	25	37	27	12	13
16	47	40	24	21	41	44	31	14	46
35	49	48	29	45	52	57	39	30	55
36	70	54	65	56	67	61	62	58	

Selection

Sort each column in ascending order

Pick out the column medians: 14 16 21 24 31 40 41 44 46 47

Find m, the median of medians

Partition A using m; $n = 49$; $|A_L| = 22$; $|A_U| = 26$;

A_L: 04 10 16 28 17 23 24 29 06 18 21 08 25 11 03 27 02 12 14 30 05 13

A_U: 35 36 32 47 49 70 33 34 40 48 54 65 45 56 41 52 67 37 44 57 61 39 62 58 46 55

Analysis

$T(n) = T(\lceil n/5 \rceil) + T(?) + cn$
What is an upper bound on the second recursive call?

As a thought exercise, imagine that the columns are permuted so that their medians are in increasing order.
Then m is in the median column
There are $\lceil \lceil n/5 \rceil /2 \rceil - 1$ columns to the left of the column of m
There are $\lfloor \lceil n/5 \rceil /2 \rfloor \leq \lceil \lceil n/5 \rceil /2 \rceil - 1$ columns to the right of the column of m; of these, all but the last are full columns
Every full column to the left of m gives at least 3 elements to A_L
Every full column to the right of m gives at least 3 elements to A_U
So, each of A_L and A_U has at least $3(\lceil \lceil n/5 \rceil /2 \rceil - 2)$ elements

Analysis

$T(n) \leq T(\lceil n/5 \rceil) + T(7n/10 + 6) + cn$

Let us prove by induction that $T(n) \leq dn$

Basis: $n \leq n_0$; sort and pick out the k-th; takes, say, time t; $T(n_0) = t = (t/n_0) * n_0$; choose $d \geq t/n_0$.

Hypothesis: $\forall m < n,\ T(m) \leq dm$

Analysis

Induction Step:

$$\begin{aligned} T(n) &\le T(\lceil n/5 \rceil) + T(7n/10 + 6) + cn \\ &\le d\lceil n/5 \rceil + 7dn/10 + 6d + cn \\ &\le d(n/5 + 1) + 7dn/10 + 6d + cn \\ &\le 9dn/10 + 7d + cn \end{aligned}$$

If $9dn/10 + 7d + cn \le dn$, then the induction holds
That is, $cn \le dn/10 - 7d$

Or, $d \ge 10cn/(n - 70)$

Choose $n_0 = 80$.
Then, for any $n \ge 80$, $10cn/(n - 70) \le 10c * 80/(80 - 70) = 80c$
Choosing $d = 80c$ ensures that the induction goes through.

Analysis

Choose $n_0 = 140$.
Then, for any $n \geq 140$,
$10cn/(n-70) \leq 10c * 140/(140-70) = 20c$
Choosing $d = 20c$ ensures that the induction goes through.

The basis dictates one choice of d; this choice increases with increasing n_0
The step dictates another; this choice decreases with increasing n_0
The final d is the larger of the two
There is an optimal choice of n_0 that minimises the final d

Analysis

The column length is hardcoded as 5 into this algorithm. Why?
It's chosen odd so that the column median is central

Then why not 3?
The induction does not go through when the column size is 3.
Redo the analysis with 3, and show that it does not work

Why not 7, 9, 11 etc.?
Induction does go through for them. The choice of 5 minimizes the final value of d. Try 7, redo the analysis, and verify.