Divide and Conquer - Part 2

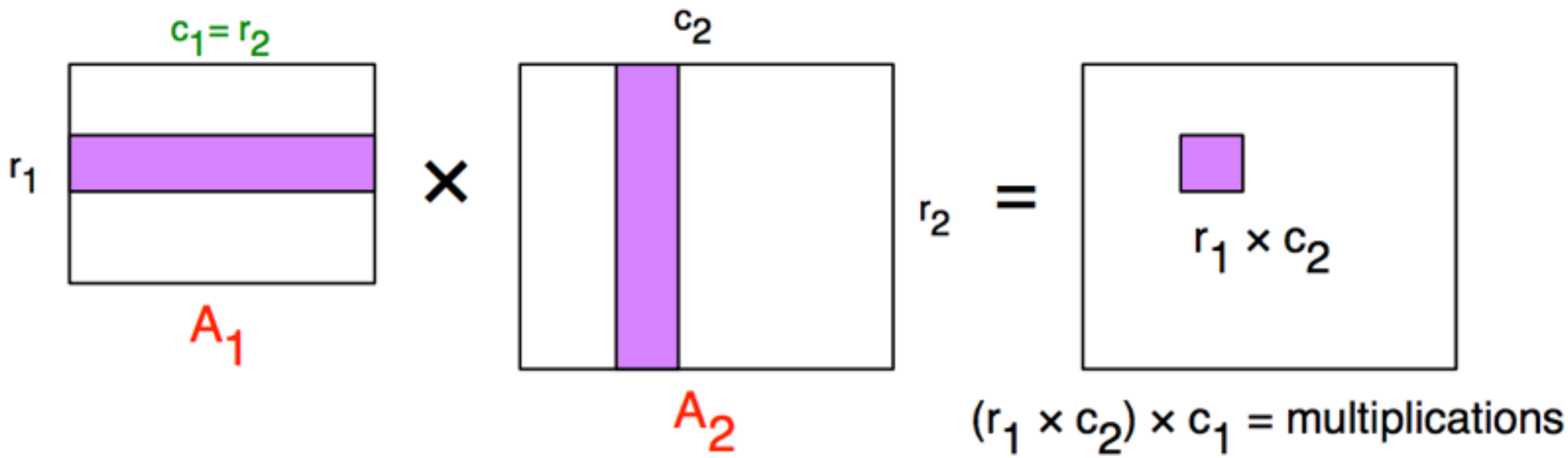
Debarka Sengupta

Content

- Strassen’s matrix multiplication
- Median finding (The Selection problem)

Strassen’s matrix multiplication

Multiplying two square matrices



If $r_1 = c_1 = r_2 = c_2 = N$, this standard approach takes $\Theta(N^3)$:

- For every row $\vec{r}$ (N of them)
- For every column $\vec{c}$ (N of them)
- Take their inner product: $r \cdot c$ using N multiplications

Multiplication - block by block

For simplicity, assume $N = 2^n$ for some n. The multiplication is:

$$N = 2^n \left\{ \underbrace{\begin{array}{|c|c|} \hline A_{11} & A_{12} \\ \hline A_{21} & A_{22} \\ \hline \end{array}}_{N = 2^n} \right. \times \begin{array}{|c|c|} \hline B_{11} & B_{12} \\ \hline B_{21} & B_{22} \\ \hline \end{array} = \begin{array}{|c|c|} \hline C_{11} & C_{12} \\ \hline C_{21} & C_{22} \\ \hline \end{array}$$

- $C_{11} = A_{11}B_{11} + A_{12}B_{21}$
- $C_{21} = A_{21}B_{11} + A_{22}B_{21}$
- $C_{12} = A_{11}B_{12} + A_{12}B_{22}$
- $C_{22} = A_{21}B_{12} + A_{22}B_{22}$

Uses 8 multiplications

A divide and conquer strategy

$$A = \begin{pmatrix} A^{11} & A^{12} \\ A^{21} & A^{22} \end{pmatrix} \quad B = \begin{pmatrix} B^{11} & B^{12} \\ B^{21} & B^{22} \end{pmatrix} \quad C = A \times B = \begin{pmatrix} C^{11} & C^{12} \\ C^{21} & C^{22} \end{pmatrix}$$

Formulas for $C^{11}, C^{12}, C^{21}, C^{22}$:

$$C^{11} = A^{11}B^{11} + A^{12}B^{21} \qquad C^{12} = A^{11}B^{12} + A^{12}B^{22}$$
$$C^{21} = A^{21}B^{11} + A^{22}B^{21} \qquad C^{22} = A^{21}B^{12} + A^{22}B^{22}$$

D&C pseudocode

MMult(A, B, n)

If $n = 1$ Output $A \times B$

Else

  Compute $A^{11}, B^{11}, \ldots, A^{22}, B^{22}$ % by computing $m = n/2$

  $X_1 \leftarrow MMult(A^{11}, B^{11}, n/2)$

  $X_2 \leftarrow MMult(A^{12}, B^{21}, n/2)$

  $X_3 \leftarrow MMult(A^{11}, B^{12}, n/2)$

  $X_4 \leftarrow MMult(A^{12}, B^{22}, n/2)$

  $X_5 \leftarrow MMult(A^{21}, B^{11}, n/2)$

  $X_6 \leftarrow MMult(A^{22}, B^{21}, n/2)$

  $X_7 \leftarrow MMult(A^{21}, B^{12}, n/2)$

  $X_8 \leftarrow MMult(A^{22}, B^{22}, n/2)$

  $C^{11} \leftarrow X_1 + X_2$

  $C^{12} \leftarrow X_3 + X_4$

  $C^{21} \leftarrow X_5 + X_6$

  $C^{22} \leftarrow X_7 + X_8$

  Output C

End If

Analysis

- 8 recursive calls

- Between lines 12-15 additions of pairs of n/2 matrices takes $n^2/4$ or $O(n^2)$ time.

$$T(n) = 8T(\frac{n}{2}) + \Theta(n^2)$$

Master method gives us …

$$T(n) = \Theta(n^{\log_2(8)}) = \Theta(n^3)$$

No improvement over the naive multiplication

Strassen's algorithm - fewer multiplication

$$\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \times \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix} = \begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix}$$

$$P_1 = (A_{11} + A_{22})(B_{11} + B_{22})$$
$$P_2 = (A_{21} + A_{22})B_{11}$$
$$P_3 = A_{11}(B_{12} - B_{22})$$
$$P_4 = A_{22}(B_{21} - B_{11})$$
$$P_5 = (A_{11} + A_{12})B_{22}$$
$$P_6 = (A_{21} - A_{11})(B_{11} + B_{12})$$
$$P_7 = (A_{12} - A_{22})(B_{21} + B_{22})$$

$$C_{11} = P_1 + P_4 - P_5 + P_7$$
$$C_{12} = P_3 + P_5$$
$$C_{21} = P_2 + P_4$$
$$C_{22} = P_1 - P_2 + P_3 + P_6$$

Uses only 7 multiplications!

D&C pseudocode

Strassen(A, B)

If $n = 1$ Output $A \times B$
Else
  Compute $A^{11}, B^{11}, \ldots, A^{22}, B^{22}$ % by computing $m = n/2$
  $P_1 \leftarrow \mathit{Strassen}(A^{11}, B^{12} - B^{22})$
  $P_2 \leftarrow \mathit{Strassen}(A^{11} + A^{12}, B^{22})$
  $P_3 \leftarrow \mathit{Strassen}(A^{21} + A^{22}, B^{11})$
  $P_4 \leftarrow \mathit{Strassen}(A^{22}, B^{21} - B^{11})$
  $P_5 \leftarrow \mathit{Strassen}(A^{11} + A^{22}, B^{11} + B^{22})$
  $P_6 \leftarrow \mathit{Strassen}(A^{12} - A^{22}, B^{21} + B^{22})$
  $P_7 \leftarrow \mathit{Strassen}(A^{11} - A^{21}, B^{11} + B^{12})$
  $C^{11} \leftarrow P_5 + P_4 - P_2 + P_6$
  $C^{12} \leftarrow P_1 + P_2$
  $C^{21} \leftarrow P_3 + P_4$
  $C^{22} \leftarrow P_1 + P_5 - P_3 - P_7$
  Output C
End If

Analysis

Recursive relation

$$T(n) = 7T(\frac{n}{2}) + \Theta(n^2)$$

Master method gives us …

$$T(n) = \Theta(n^{\log_2(7)}) = \Theta(n^{2.8})$$

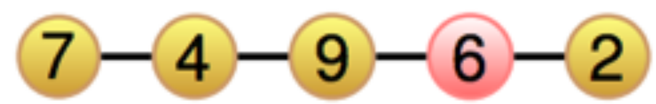
Median finding (The Selection problem)

Selection problem

The selection problem:

- given an integer k and a list $x_1, \ldots, x_n$ of n elements
- find the k-th smallest element in the list

Example: the 3rd smallest element of the following list is 6



An $O(n \cdot \log n)$ solution:

- sort the list ($O(n \cdot \log n)$)
- pick the k-th element of the sorted list ($O(1)$)

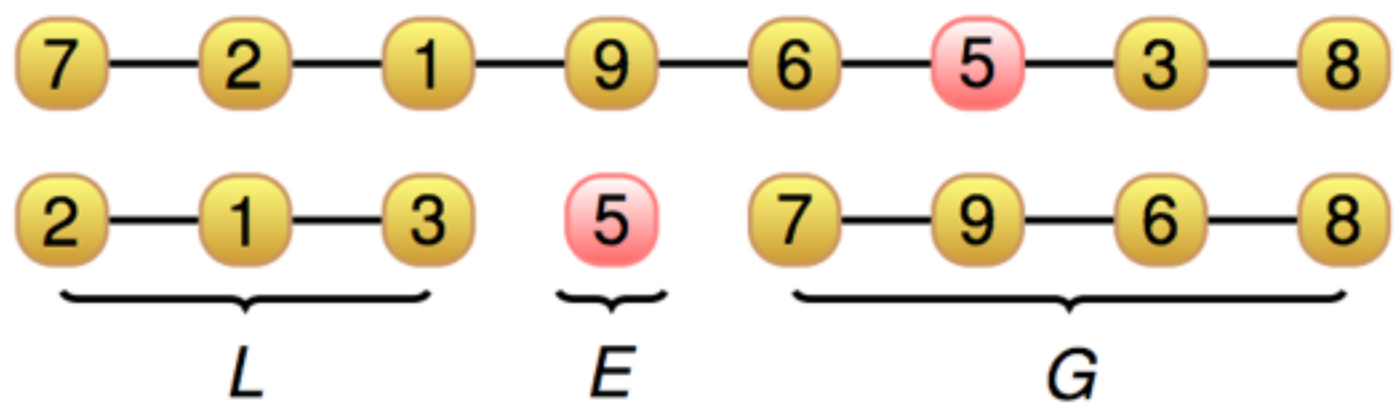
Can we find the k-th smallest element faster?

Quick-Select

Quick-select of the k-th smallest element in the list S:

Decrease and conquer

- based on **prune-and-search** paradigm
- Prune: pick random element x (pivot) from S and split S in:
  - L elements $< x$, E elements $== x$, G elements $> x$



  - partitioning into L, E and G works precisely as for quick-sort
- Search:
  - if $k \leq |L|$ then return quickSelect(k, L)
  - if $|L| < k \leq |L| + |E|$ then return x
  - if $k > |L| + |E|$ then return quickSelect$(k - |L| - |E|, G)$

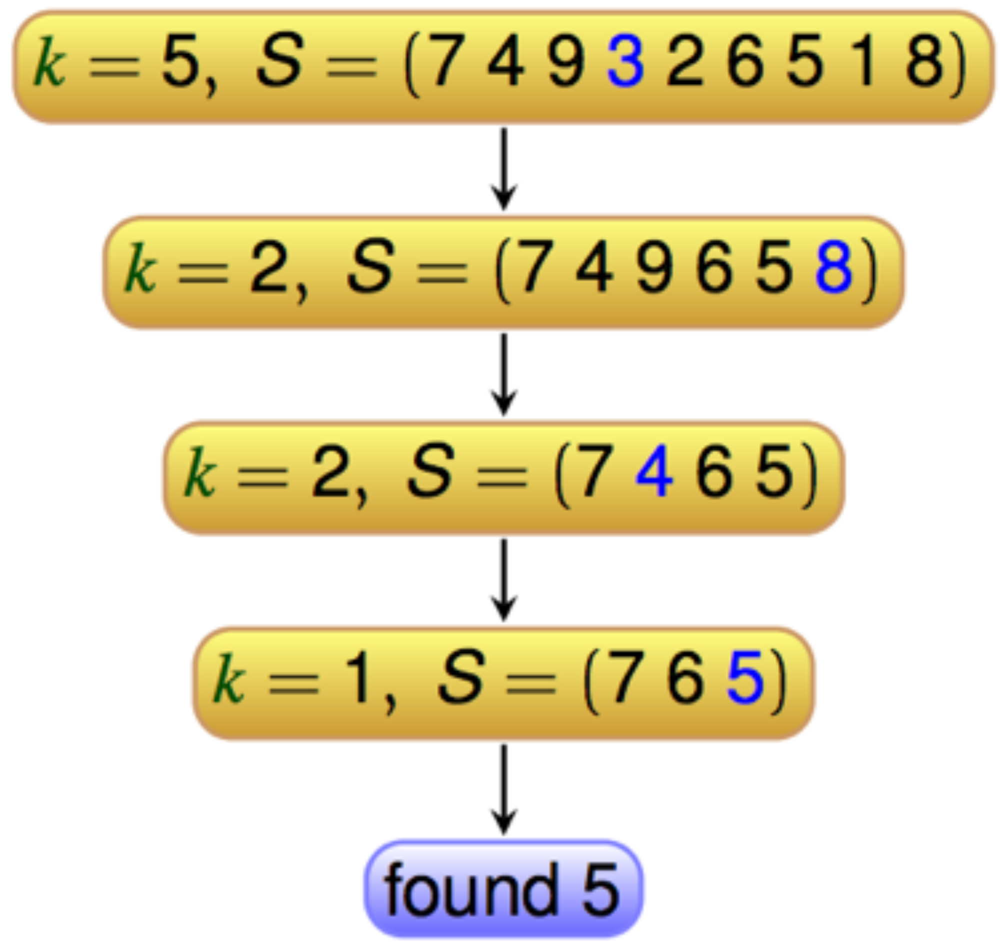
Not to forget

- For median, k = ceiling of (N/2)

Example

Quick-select can be displayed by a sequence of nodes:

- each node represents recursive call and stores: k, the sequence, and the pivot element



Analysis

- The worst case running time is $O(n^2)$ - It happens when the pivot is always the minimal or maximal element

- The expected running time if $O(n)$. Let's see how.

Analysis

- Rank of the randomly decided pivot is uniformly distributed between 0 and N-1, where N is the length of the data

- This means there's at least a 1/2 chance that it's between N/4 and 3N/4.

- That means that at least half the time, we get a pivot that reduces the problem to no more than 3/4 of the previous problem size.

Analysis

- On every iteration running the input N items by the pivot takes O(N) time
- Based on the above, we expect the problem to be reduced to no more than 3/4 of its size on an average after every second of these O(N) passes.
- We therefore get a recurrence relationship of

$$T(N) = T(3N/4) + 2 * O(N)$$

$$= T(3N/4) + O(N)$$

Solving the relation

$$T(N) = cN + T(3/4N)$$

$$= cN + c(3/4N) + T((3/4)^2 N) \ \ldots$$

$$= cN + c(3/4N) + c((3/4)^2 N) + \ldots$$

$$= cN(1 + 3/4 + (3/4)^2 + \ldots)$$

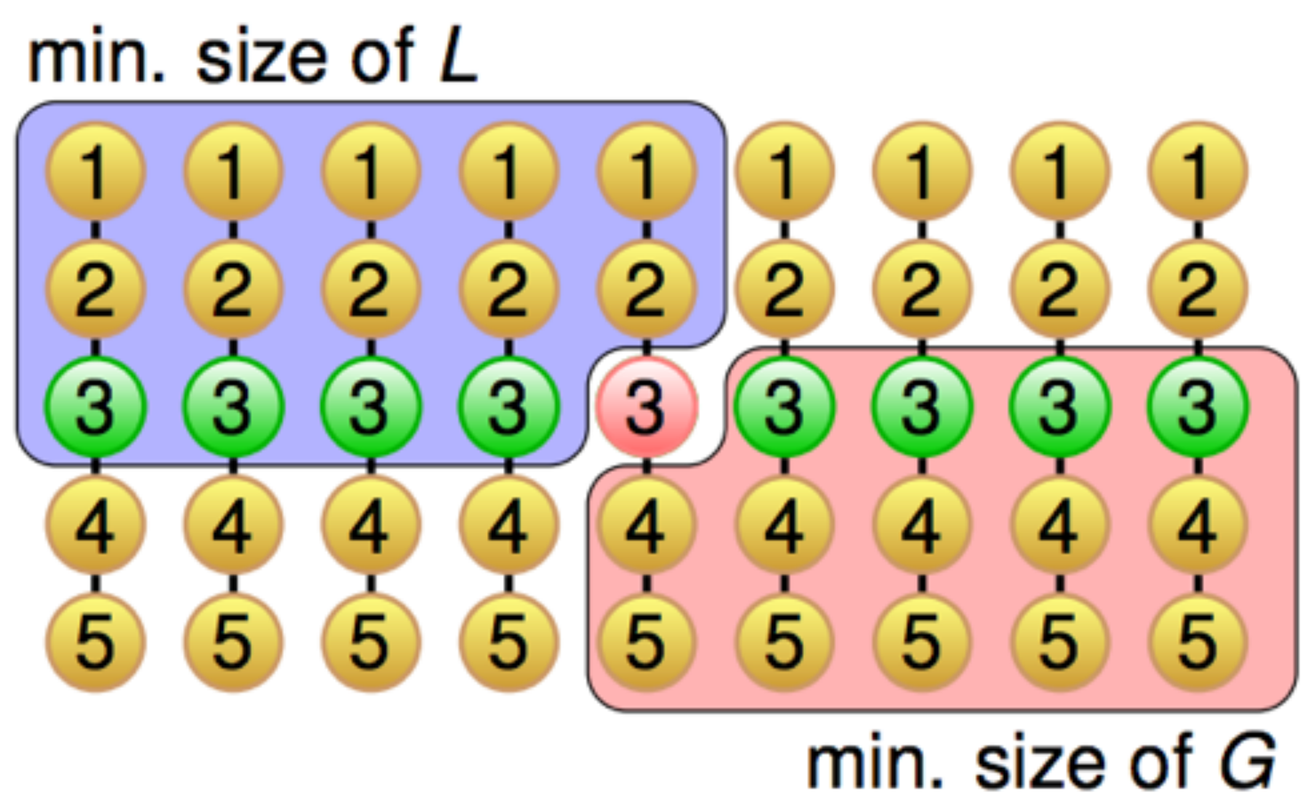
$= O(N)$ [Since the geometric series sums to a constant]

A plan to improve the worst case

We can do selection in $O(n)$ worst-case time.

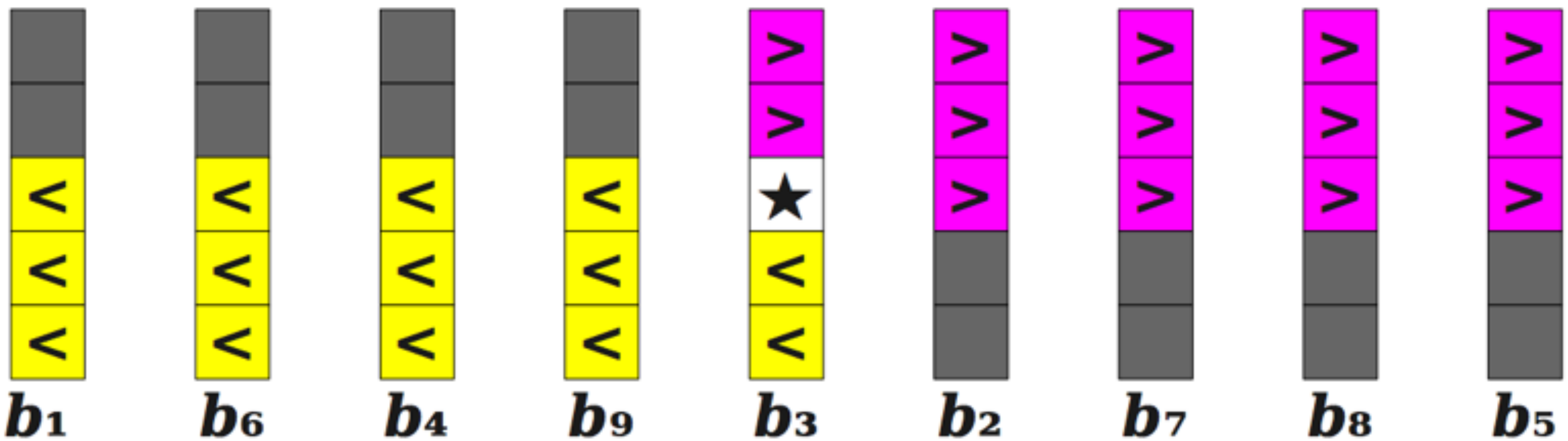
Idea: recursively use select itself to find a good pivot:

- divide S into $n/5$ sets of 5 elements
- find a median in each set (baby median)
- recursively use select to find the median of the medians



The minimal size of L and G is $0.3 \cdot n$.

To understand it better



- The median-of-medians (★) is larger than 3/5 of the elements from (roughly) the first half of the blocks.
- ★ larger than about 3/10 of the total elements.
- ★ is smaller than about 3/10 of the total elements.
- Guarantees a 30% / 70% split.

A numerical example

2	18	110	0	3
7	41	110	3	4
99	110	111	115	116
106	112	112	120	129
120	113	115	190	6000

Pseudocode

```
select(L,k)
{
if (L has 10 or fewer elements)
{
    sort L
    return the element in the kth position
}

partition L into subsets S[i] of five elements each
    (there will be n/5 subsets total).

for (i = 1 to n/5) do
    x[i] = select(S[i],3)

M = select({x[i]}, n/10)

partition L into L1<M, L2=M, L3>M
if (k <= length(L1))
    return select(L1,k)
else if (k > length(L1)+length(L2))
    return select(L3,k-length(L1)-length(L2))
else return M
}
```

Analysis

- In the worst possible case, every element in the blue rectangle (top left) will be less than or equal to our pivot.
- The blue box contains $3/5^{th}$ of the 1/2 or of the elements $3/10^{th}$ of the elements
- So every pivot chosen in this manner would have (roughly) at least $3/10^{th}$ of the elements lesser and $3/10^{th}$ of the elements greater than it

At each recursive step

- At each step:
  - $O(N)$ work to partition N elements into $N/5$ blocks of size 5 (each)
  - Sub problem $7/10^{th}$ of the size of the original problem
  - Solve 1 subproblem $1/5^{th}$ the size of the original to compute the median of medians
- This yields the following.

$$\mathbf{T(n) = T(n/5) + T(7n/10) + O(N)}$$

Materials used

- https://rcoh.me/posts/linear-time-median-finding/
- https://web.stanford.edu/class/archive/cs/cs161/cs161.1138/lectures/08/Small08.pdf
- http://www.few.vu.nl/~tcs/ds/lecture9.pdf
- http://people.seas.harvard.edu/~cs125/fall16/lec4.pdf
- https://www.cs.mcgill.ca/~pnguyen/251F09/matrix-mult.pdf
- https://www.cs.cmu.edu/~ckingsf/class/02713/lectures/lec14-strassen.pdf
- https://www.ics.uci.edu/~eppstein/161/960130.html

End