# 11.4.6
# Epilogue: On selection in linear time

# Summary: Selection in linear time

**Theorem**

*The algorithm* **select**$(A[1 .. n], k)$ *computes in* $O(n)$ *deterministic time the* $k$*th smallest element in* $A$.

On the other hand, we have:

**Lemma**

*The algorithm* **QuickSelect**$(A[1 .. n], k)$ *computes the* $k$*th smallest element in* $A$. *The running time of* **QuickSelect** *is* $\Theta(n^2)$ *in the worst case.*

# Questions to ponder

1. Why did we choose lists of size $5$? Will lists of size $3$ work?
2. Write a recurrence to analyze the algorithm's running time if we choose a list of size $k$.

# Median of Medians Algorithm

Due to:
M. Blum, R. Floyd, D. Knuth, V. Pratt, R. Rivest, and R. Tarjan.
"Time bounds for selection".
Journal of Computer System Sciences (JCSS), 1973.

How many Turing Award winners in the author list?
All except Vaughn Pratt!

# Median of Medians Algorithm

Due to:
M. Blum, R. Floyd, D. Knuth, V. Pratt, R. Rivest, and R. Tarjan.
"Time bounds for selection".
Journal of Computer System Sciences (JCSS), 1973.

How many Turing Award winners in the author list?
All except Vaughn Pratt!

# Median of Medians Algorithm

Due to:
M. Blum, R. Floyd, D. Knuth, V. Pratt, R. Rivest, and R. Tarjan.
"Time bounds for selection".
Journal of Computer System Sciences (JCSS), 1973.

How many Turing Award winners in the author list?
All except Vaughn Pratt!

# Takeaway Points

1. Recursion tree method and guess and verify are the most reliable methods to analyze recursions in algorithms.
2. Recursive algorithms naturally lead to recurrences.
3. Some times one can look for certain type of recursive algorithms (reverse engineering) by understanding recurrences and their behavior.