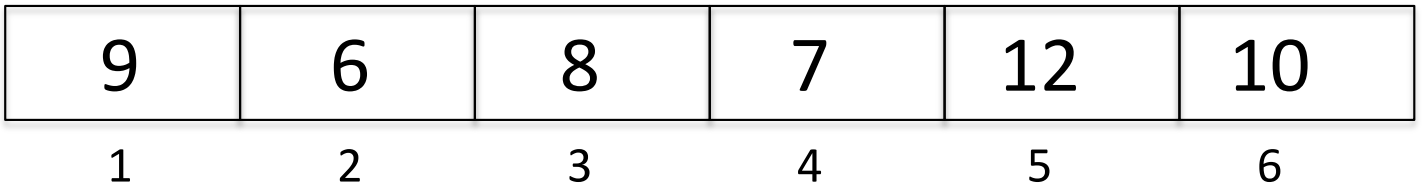
Using divide and conquer

9/18/24

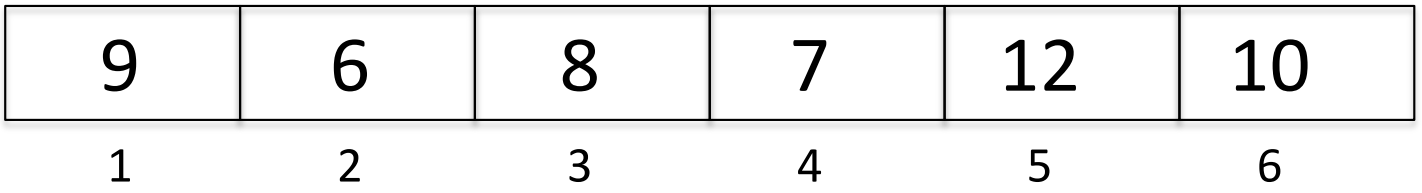
Using Divide and Conquer: Maximum profit

- Maximize profit from one purchase and sale of stock given an array of prices over time:



Using Divide and Conquer: Maximum profit

- Maximize profit from one purchase and sale of stock given an array of prices over time:



- Is the max profit always from buying at the lowest price?
  A. Yes
  B. No

Using Divide and Conquer: Maximum profit

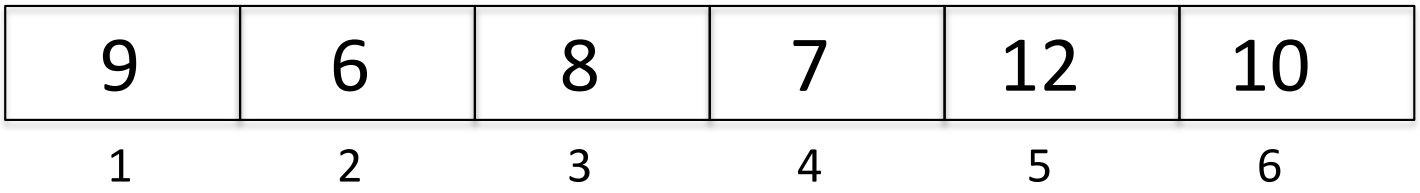
- Maximize profit from one purchase and sale of stock given an array of prices over time:

9	6	8	7	12	10
1	2	3	4	5	6

- Is the max profit always from buying at the lowest price?
  A. Yes
  B. No

Using Divide and Conquer: Maximum profit

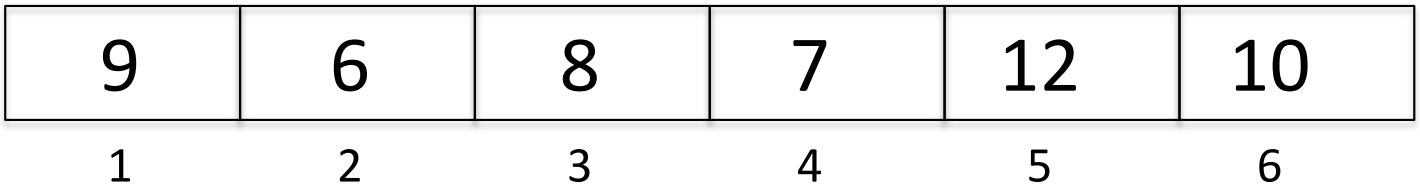
- Maximize profit from one purchase and sale of stock given an array of prices over time:



- D&C algorithm:
  - Split array into halves
  - Determine max profit, max price, and min price for each half
  - Combine these to compute own return values

Using Divide and Conquer: Maximum profit

- Maximize profit from one purchase and sale of stock given an array of prices over time:



- What recurrence describes the running time of our algorithm?

A. $T(n) = 2T(n/2) + 1$

B. $T(n) = 2T(n/2) + \log n$

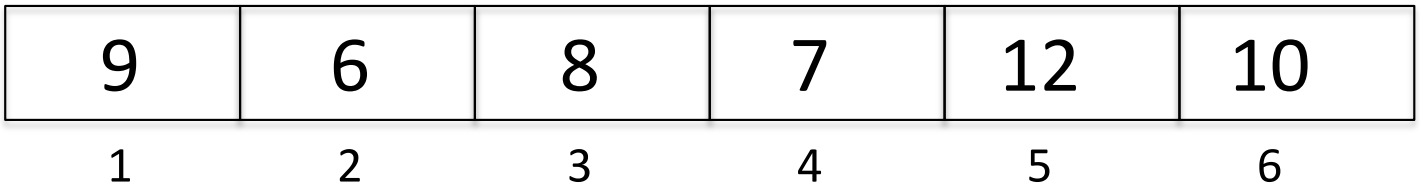
C. $T(n) = 2T(n/2) + n$

D. $T(n) = 2T(n/2) + n^2$

E. None of the above

Using Divide and Conquer: Maximum profit

- Maximize profit from one purchase and sale of stock given an array of prices over time:



- What recurrence describes the running time of our algorithm?

A. $T(n) = 2T(n/2) + 1$

B. $T(n) = 2T(n/2) + \log n$

C. $T(n) = 2T(n/2) + n$

D. $T(n) = 2T(n/2) + n^2$

E. None of the above

In the book: Transform problem to finding maximum subarray

- Maximize profit from one purchase and sale of stock given an array of prices over time:

9	6	8	7	12	10
1	2	3	4	5	6

- Find the contiguous subarray with largest sum

-3	2	-1	5	-2
2	3	4	5	6

Next problem: Multiplying large integers

$$
\begin{array}{rr}
 & 782934728937492347982378942 \\
\times & 378923478397498327483279 2 \\
\hline
\end{array}
$$

What is the running time of the grade-school algorithm (n-digit integers)?

A. O(1)

B. O(n)

C. O(n log n)

D. $O(n^2)$

E. None of the above

What is the running time of the grade-school algorithm (n-digit integers)?

A. O(1)

B. O(n)

C. O(n log n)

D. $O(n^2)$

E. None of the above