

2019/02/24.

```
const fn = (kTimes, prices) => {  
  if (!kTimes || !prices.length) {  
    return 0;  
  }  
  const n = prices.length;  
  if (kTimes >= n/2) {  
    return greedy(prices);  
  }  
  return dpFn(kTimes, prices);  
}
```

```
function dpFn(kTimes, prices) {  
  const n = prices.length;  
  const dp = new Array(n).fill(0);  
  let min = prices[0];  
  for (let i = 0; i < n; i++) {  
    min = Math.min(min, prices[i]);  
    dp[i] = prices[i] - min;  
    dp[i] = Math.max(dp[i-1], prices[i] - min);  
  }
```

```

    for (let k = 2; k <= kTimes; k++) {
        let maxDiff = dp[0] - prices[0];
        for (let i = 0; i <= n; i++) {
            get 'diff' before
            'dp[i]' changed.
            const diff = dp[i] - prices[i];
            dp[i] = Math.max(
                maxDiff + prices[i],
                dp[i - 1],
            );
            maxDiff = Math.max(maxDiff, diff);
        }
    }
    return dp[n - 1];
}

```

```

function greedy(prices) {
    let output = 0;
    for (let i = 1; i < prices.length; i++) {
        if (prices[i] > prices[i - 1]) {
            output += prices[i] - prices[i - 1];
        }
    }
    return output;
}

```

[3, 2, 6, 5, 0, 3], k = 3

		$\uparrow$ $\bar{j}$	$\uparrow$ $\bar{i}$
k			
0			
1		$\bigcirc_{\bar{j}}$	
2			$\bigcirc_{\bar{i}}$
3			

$$dp[k][i] = \text{Max} \left\{ \begin{array}{l} dp[k-1][j] + p[i] - p[j] \\ \quad \bar{j} \text{ from } 0 \text{ to } i-1 \\ dp[k][i-1] \end{array} \right\}$$

$$dp[i] = \text{Max} \left\{ \begin{array}{l} dp[j] + p[i] - p[j], \\ \quad \bar{j} \text{ from } 0 \text{ to } i-1 \\ dp[i-1] \end{array} \right\}$$

$$= \text{Max} \left\{ \begin{array}{l} \textcircled{1} \\ \underline{dp[j] - p[j] + p[i]} \\ \quad \bar{j} \text{ from } 0 \text{ to } i-1 \\ dp[i-1] \end{array} \right\}$$

① this is the max value of  $dp[j] - p[j]$ . for  $\bar{j} < i$ .

so we need only 1 variable to keep tracking this without looping  $\bar{j}$  from 0 to  $i-1$ .

$dp[i] = \text{Math.max}(\text{maxDiff} + p[i], dp[i-1]);$

$\text{maxDiff} = dp[i] - p[i]$  ←

// update maxDiff for next i.