

8. High Frequency

九章算法IT求职面试培训 第8章

www.ninechapter.com

Outline

1. Single Number I, II, III
2. Majority Number I, II, III
3. Best Time to Buy and Sale Stock I, II, II
4. Subarray I, II, III, IV
5. 2-Sum, 3-Sum, 4-Sum, k-Sum, 3-Sum Closest
6. Quick Questions
7. Partition Array

Single Number

<http://lintcode.com/zh-cn/problem/single-number/>

Single Number II

<http://oj.leetcode.com/problems/single-number-ii/>

Single Number III

Given an array of integers, every element appears twice except for two. Find the two singles.

Majority Number

<http://lintcode.com/en/problem/majority-number/>

Majority Number II

<http://lintcode.com/en/problem/majority-number-ii/>

Majority Number II

<http://lintcode.com/en/problem/majority-number-iii/>

Majority Number III

<http://lintcode.com/en/problem/majority-number-iii/>

Best Time to Buy and Sell Stock

<http://oj.leetcode.com/problems/best-time-to-buy-and-sell-stock/>

Best Time to Buy and Sell Stock II

<http://oj.leetcode.com/problems/best-time-to-buy-and-sell-stock-ii/>

Best Time to Buy and Sell Stock III

<http://oj.leetcode.com/problems/best-time-to-buy-and-sell-stock-iii/>

Best Time to Buy and Sell Stock IV

k transactions

state: $f[i][j]$ 表示前 i 天进行 j 次交易, 能够获得的最大收益

function: $f[i][j] = \max\{f[x][j-1] + \text{profit}(x+1, i)\}$

answer: $f[n][k]$

intialize: $f[i][0] = 0, f[0][i] = -\text{MAXINT} (i > 0)$

Subarray

<http://lintcode.com/en/problem/maximum-subarray/>

<http://lintcode.com/en/problem/maximum-subarray-ii/>

<http://lintcode.com/en/problem/maximum-subarray-iii/>

Subarray II

<http://lintcode.com/en/problem/minimum-subarray/>

Subarray III

<http://lintcode.com/en/problem/maximum-subarray-difference/>

Subarray IV

1. Find the subarray which sum equals to zero.
time: $O(n)$ space: $O(n)$
2. Find the subarray which sum is closest to zero.
time: $O(n \log n)$ space: $O(n)$

2-Sum

<http://lintcode.com/en/problem/2-sum/>

3-Sum

<http://lintcode.com/en/problem/3-sum/>

3-Sum Closest

<http://lintcode.com/en/problem/3-sum-closest/>

4-Sum

<http://lintcode.com/en/problem/4-sum/>

k-Sum

?

k sum II

给n个 **互不相同的数**, 让你取k个数, 问这k个数之和是target的, **有多少种方案**

state: $f[n][k][target]$ 前n个数, 取k个数, 组成和为target的方案有多少个

function: $f[n][k][target] = f[n-1][k-1][target-a[n]] + f[n-1][k][target]$

Quick Questions

- **Power(x, n)**
 - $x^n = (x^{(n/2)})^2$ 快速幂 $\rightarrow O(\log n)$
- **Sqrt(x)**
 - Magic Number 0x5f3759df
- **Trailing Number of zeros in n!**
- **O(1) Check Power of 2**
 - $(x-1) \& x == 0 \rightarrow x$ 是 2 的某次幂

Partition Array

Given an array "a" of integers and an int "k",
Partition the array (i.e move the elements in
"a") such that

- All elements $< k$ are moved to the left
- All elements $\geq k$ are moved to the right

Return the partitioning Index, i.e the first
index "i" $a[i] \geq k$.

Partition Array

<http://lintcode.com/en/problem/partition-array/>

Sort Letters by Case

<http://lintcode.com/en/problem/sort-letters-by-case/>

Sort Colors

<http://oj.leetcode.com/problems/sort-colors/>

Interleaving Negative & Positive numbers

$[1, 2, -1, 3, 4, -5, -6] \Rightarrow [1, -1, 2, -5, 3, -6, 4]$