

Max Howell

@mxcl

Swift. Open Source. Futurism. Creator of Homebrew.

Savannah, GA



Max Howell

@mxcl

关注

Google: 90% of our engineers use the software you wrote (Homebrew), but you can't invert a binary tree on a whiteboard so fuck off.

上午10:07 - 2015年6月10日

7,305 转推 10,384 喜欢



518

7.3千

1万



RoToЯ @the_real_rotor · 2015年6月10日

回复 @mxcl

@mxcl @tef I received earnest advice on how to study for a google interview and decided if I can cram for an interview I don't want the job

3

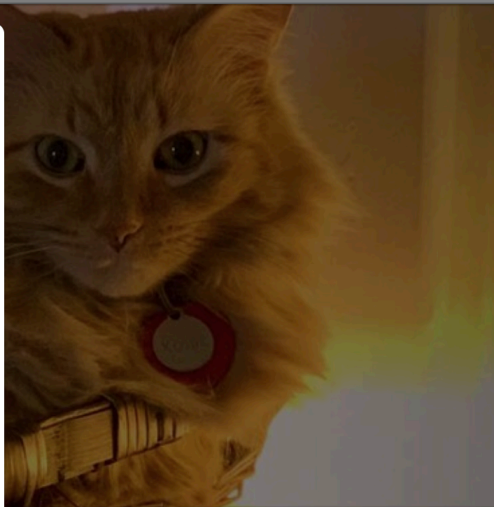
11

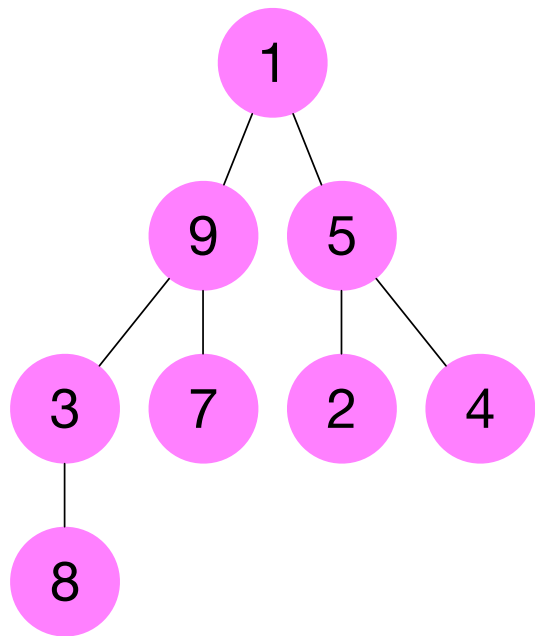
92



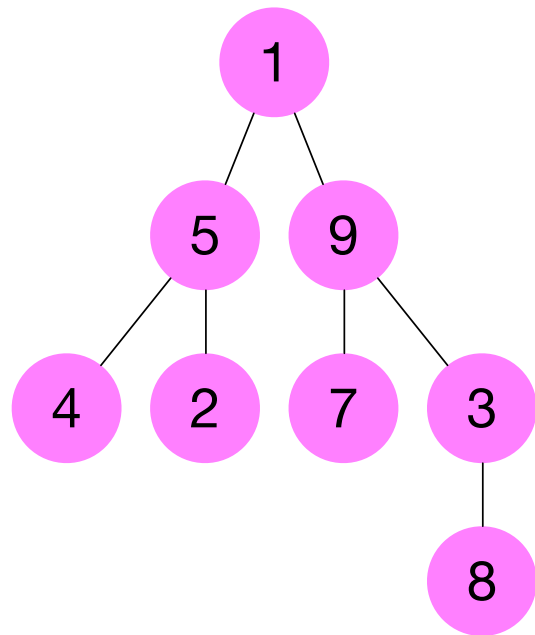
Erik Rose @ErikRose · 2015年6月11日

.@the_real_rotor @mxcl @tef @rnewman OTOH, reading The Algorithm Design Manual is its own reward.





Max Howell @mxcl · 9h
Google: 90% of our engineers use the software you wrote (Homebrew), but you can't invert a binary tree on a whiteboard so fuck off.



常见-反转二叉树

```
function reverseBTree(node) {
```

```
  if(!node) {
```

```
    return
```

```
  }
```

```
  const tmp = node.left
```

```
  node.left = node.right
```

```
  node.right = tmp
```

```
  reverseBTree(node.left)
```

```
  reverseBTree(node.right)
```

```
}
```

常见面试题-解析URL

写一个程序，解析下面的queryString，返回对象

a.name=ramroll&a.dress&x=1&y=



今日头条



各种问题

- `a[name]=枭兽&a[gender]=1`
- `color=Deep%20Blue`
- `ids[0]=1&ids[1]=2&ids[2]=3`

为什么考反转二叉树

- 过滤掉理解不了问题的人 - 交流障碍
- 过滤掉写递归要写一个小时的人 - 手慢
- 过滤掉二叉树都不知道的人 - 工作范围窄
- 过滤掉思考深度不够的人 - BUG多



今日头条 N个数字和为M的问题

给定一个不重复的正整数集合，从中取N个数字，使得他们的和为M，写一个函数，求这个N个数字。如有多个，只需要返回一个。

```
function sumN(A, n, m)
```

解法-决策树 递归

```
function sumN(A, n, m, i=0, decisions = []) {  
  if(m === 0){  
    return decisions  
  }  
  if(i === A.length || n === 0){  
    return null  
  }  
  return sumN(A, n-1, m-A[i], i+1, decisions.concat(A[i]))  
    || sumN(A, n, m, i+1, decisions)  
}
```


解法-利用位运算

对于A=[1,2,3,4,5] 二进制数10011代表[1,4,5]

```
function sumByBinaryCode(A, code) {  
  const max = 1 << A.length  
  const p = []  
  let sum = 0  
  for(let i = 0; i < A.length; i++) {  
    if (code & (1 << i)) {  
      sum += A[i]  
      p.push(A[i])  
    }  
  }  
  return {sum, p}  
}
```

```
sumByBinaryCode([1,2,3,4], 0b1111)  
▼ {sum: 10, p: Array(4)} ⓘ  
  ► p: (4) [1, 2, 3, 4]  
    sum: 10  
  ► __proto__: Object
```

解法-遍历Binary Number

```
function sumN(A, n, m) {  
    const max = 1 << A.length //  
max=2^(A.length)  
    for(let i = 0; i < max; i++) {  
        const {sum, p} = sumByBinaryCode(A, i)  
        if (sum === m) {  
            return p  
        }  
    }  
    return null  
}
```

误区

- 容易和「0-1背包问题」弄混。举例：给定一个不重复的整数数组，从中取N个数字，使得他们的和最大？动态规划(0-1背包)
- 一个集合，求和为M的子集，是一个NP-完全

时间标准

- 没写出 —— 不行
- 30分钟写出 —— 弱
- 20分钟写出 —— 还行
- 10分钟写出 —— 优秀
- 秒写 —— Strong Agree

知识标准

- 不知道解法 —— 不行
- 暴力解写得很长，不太会写，没有套路 —— 弱
- 暴力解写得够精简 —— 还行
- 给出一种方案，说出优化解，知道时间复杂度 —— 优秀
- 能说出是NP-Hard问题，知道0-1背包 —— Strong Agree

为什么考算法？

我需要他写代码快！

我希望他bug少！

我希望他工作范围广！



切图ing.....

淘宝-25匹马5个跑道，求最快的5匹，需要多少次？



关联问题 – 淘宝面试题

- 硬盘上有25个1T的文件,每个1T的文件都已经排好序了,但内存只有1G,里面都是数字,如何取出前5T大的数字?

前3个最小值一定在第一排

10	16	71	2	15	18	8	11	13
72	76	81	19	88	89	15	20	33
...				

BUFFER

--	--	--

10	16	71
----	----	----

72	76	81
----	----	----

...		
-----	--	--

2	15	18
---	----	----

19	88	89
----	----	----

...		
-----	--	--

8	11	13
---	----	----

15	20	33
----	----	----

...		
-----	--	--

BUFFER

2	8	10
---	---	----



10	16	71
----	----	----

72	76	81
----	----	----

...		
-----	--	--

2	15	18
---	----	----

19	88	89
----	----	----

...		
-----	--	--

8	11	13
---	----	----

15	20	33
----	----	----

...		
-----	--	--

BUFFER

11	13	
----	----	--

10	16	71
----	----	----

72	76	81
----	----	----

...		
-----	--	--

2	15	18
---	----	----

19	88	89
----	----	----

...		
-----	--	--

15	20	33
----	----	----

...		
-----	--	--

BUFFER

11	13	
----	----	--

10	16	71
----	----	----

72	76	81
----	----	----

...		
-----	--	--

2	15	18
---	----	----

19	88	89
----	----	----

...		
-----	--	--

15	20	33
----	----	----

...		
-----	--	--

BUFFER

11	13	15
----	----	----



10	16	71
----	----	----

72	76	81
----	----	----

...		
-----	--	--

2	15	18
---	----	----

19	88	89
----	----	----

...		
-----	--	--

15	20	33
----	----	----

...		
-----	--	--

BUFFER

--	--	--

10	16	71
----	----	----

72	76	81
----	----	----

...		
-----	--	--

2	15	18
---	----	----

19	88	89
----	----	----

...		
-----	--	--

15	20	33
----	----	----

...		
-----	--	--

BUFFER

15	16	18
----	----	----



10	16	71
----	----	----

72	76	81
----	----	----

...		
-----	--	--

19	88	89
----	----	----

...		
-----	--	--

15	20	33
----	----	----

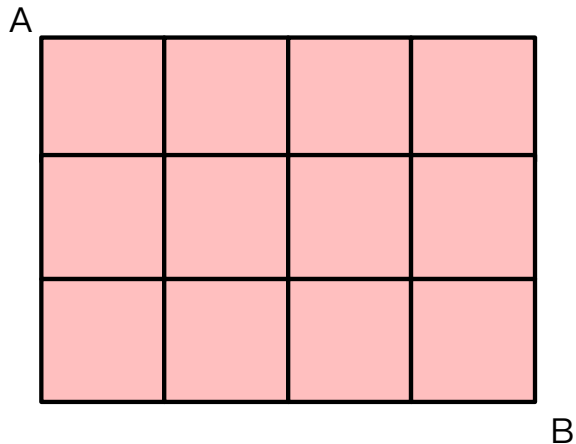
...		
-----	--	--

BUFFER

--	--	--

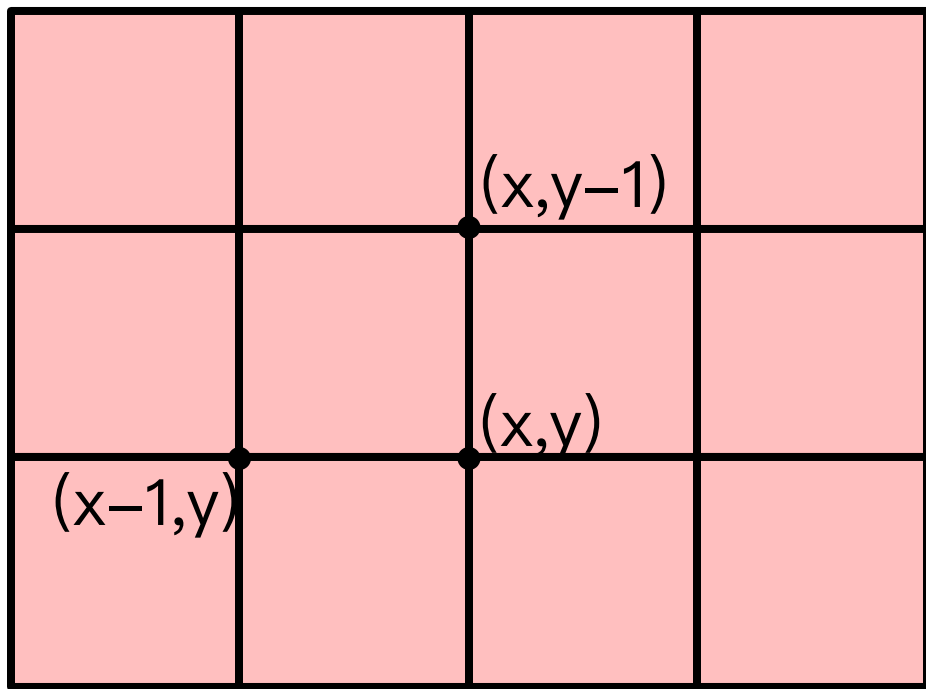


美团 数网格中的路径



- 小虫子从A到B，只能沿着网格走，每次只能向右或向下。求有多少种走法？

解法-动态规划



$$f(x, y) = \begin{cases} f(x-1, y) + f(x, y-1) + 1, & x > 0, y > 0 \\ 0, & x = 0, y = 0 \\ f(x-1, y), & x > 1, y = 0 \\ f(x, y-1), & y > 1, x = 0 \end{cases}$$

解法：排列组合

- 相当于向左走4步（4个红球）和向下走3步（3个白球），求全排列

- $\frac{A_7^7}{A_4^4 A_3^3} = 35$

为什么考算法？

我希望他真的了解性能！

我希望他足够聪明！

回到我们的问题，前端工程师算法应该学到什么程度？

- Leetcode easy难度在10min内可以给出解（业务很熟练，开发速度可以）
- Leetcode medium难度可以经过思考接触（可以做很多事，造轮子无压力）
- Leetcode hard难度的可以很快看懂答案（给我时间Google就行，我都能搞定！）

如何学习

- 不要占用上班时间刷题（请重要的事情说3遍，否则得不偿失）
- 不要再睡前刷题（卡住了就睡不着，影响第2天工作）
- 节省时间，坚持训练([leetcode](#))
- 补充理论知识([geeksforgeeks](#),算法导论等等)