

# MSRA intern 面经

## 一面

开场自我介绍，然后追问了其中一个项目，讲了大概30min。然后问了点机器学习八股：

- 介绍一下drop out和batch normalization：公式，作用，在训练和测试的时候的区别？
  - 背一下公式
  - 作用是对抗过拟合
  - 训练时生效，测试时不生效
  - drop out在训练时要乘一个系数，系数为drop out率的相反数，使得预测结果期望不变
- 会用tensorflow/pytorch吗？说一下他们的原理
  - 介绍了一下计算图是怎么工作的

中间答得不是很好，面试官说希望你回去以后补一下基础知识，看看CS231n什么的。

## 二面

二面隔了一个多礼拜，本来以为已经挂了，没想到被捞起来了。开场还是自我介绍+项目30min，然后开始切题。

## 切题

### 4-sum

给一个数组，让你找出四个数，和为给定的值S,判断是否存在这样的解。

### min width

给n个数组，每个数组里有k个元素，他们单调增，比如：

```
[1,3,5,7]
[2,4,6,8]
[3,100,101,102]
```

希望你找到这样一组a,b，使得任意一个数组里至少一个元素在 [a,b] 区间中，问区间长度最小是多少，即最小化

## binary search

给你一个单调增数组，从中间切开，然后把后面那段搬到前面，最后得到一个形如：

```
[5,6,7, 1,2,3,4]
```

的数组，现在要在里面搜某个特定元素。

## Solution

### 4 SUM

类似 2-sum，一边维护hash，一边去凑。

两重循环，第一重枚举【第三个被选择的数字】，  
里面先在当前位置之前循环一次，更新hash表，维护前两个数字可能出现的和；  
然后再在当前位置之后循环一次，枚举【第四个被选择的数字】，和S做差，判断是否在hash表中。

```
Input: n, a[..], s
Output: true/false
hash = {}
for i = 3..n-1:
    for j = 1..i-2:
        hash.append(a[i-1]+a[j])
    for j = i+1..n:
        tgt = s-a[i]-a[j]
        if tgt in hash:
            return 1
return 0
```

## min width

直接对他们归并排序，维护优先队列的时候，记录优先队列里的最大最小值，有很多种实现方法。

## binary search

还是二分，只不过判断条件稍微改一下，要进行两次比较：和当前位置左右的元素比，和最后一个元素比，分三段来确定接下来的搜索区间。

## 三面

还是自我介绍+项目，讲的比较久，剩下时间就问了一个简单算法题：非递归版本后续遍历