SAP面试题

1. Design an Employee class
https://www.zybuluo.com/mdeditor#1450643

2. Add a certain feature to a paint program.
https://www.bbsmax.com/A/MAzAx8qd9p/

3. OOP. Linked Lists. https://cs1331.gatech.edu/slides/linked-lists.pdf

4. Calculator (Leetcode 227)
https://blog.csdn.net/tiandixuanwuliang/article/details/78298089

5. implement a hash
https://stackoverflow.com/questions/5407421/design-a-hashtable

6. Implement a cache in any language you choose (Leetcode 146)

7. Design classes for a car and truck

```
class Vehicles {
    private String brand;// 商标
    private String color;// 颜色
    public Vehicles(String brand, String color) {
        this.brand = brand;
        this.color = color;
    }
    public void run() {
        System.out.println("启动中...");
    }
    public void showInfo() {
        System.out.println("商标:" + brand + ";颜色:" + color + "\n");
    }
}
```

```java
class Car extends Vehicles {// 小汽车
    private int seats;
    public Car(String brand, String color) {
        super(brand, color);
    }
    public void setSeate(int seats) {
        this.seats = seats;
    }
    public int getSeats() {
        return seats;
    }
    public void showCar() {
        System.out.println("座位数:" + seats + "张");
    }
}

class Truck extends Vehicles {// 继承商标-颜色
    // 卡车
    private double load;// 成员变量
    public Truck(String brand, String color) { // 构造方法
        super(brand, color);
    }
    public double getLoad() {// 成员变量的get
        return load;
    }
    public void setLoad(double load) {// 成员变量的set
        this.load = load;
    }
    public void showTruck() {// 成员方法
        System.out.println("载重:" + load + "KG");
    }
```

Java

1. the difference between "interface" and "abstract class"
2. What is a constructor?

Coding

1. Code Review based on Nodes/LinkedList concept

2. Create a function to reverse a string

```
class Solution(object):
    def reverseString(self, s):
        """
        :type s: str
        :rtype: str
        """
        if not s:
            return s
        l, r = 0, len(s)-1
        while l<r:
            s[l], s[r] = s[r], s[l]
            l += 1
            r -= 1
        return s

    def reverseString(self, s):
        return s[::-1]
```

3. FizzBuzz

```python
class Solution:
    def fizzBuzz(self, n):
        """
        :type n: int
        :rtype: List[str]
        """
        res = []
        for i in range(1, n+1):
            temp = ''
            if i % 3 == 0:
                temp += 'Fizz'
            if i % 5 == 0:
                temp += 'Buzz'
            if not temp:
                temp = str(i)
            res.append(temp)
        return res
```

4. Check if a string is a palindrome

learning_solution.py

```python
class Solution:
    def isPalindrome(self, s: str) -> bool:
        alphanumericS = [c for c in s.lower() if c.isalnum()]
        return alphanumericS == alphanumericS[::-1]
```

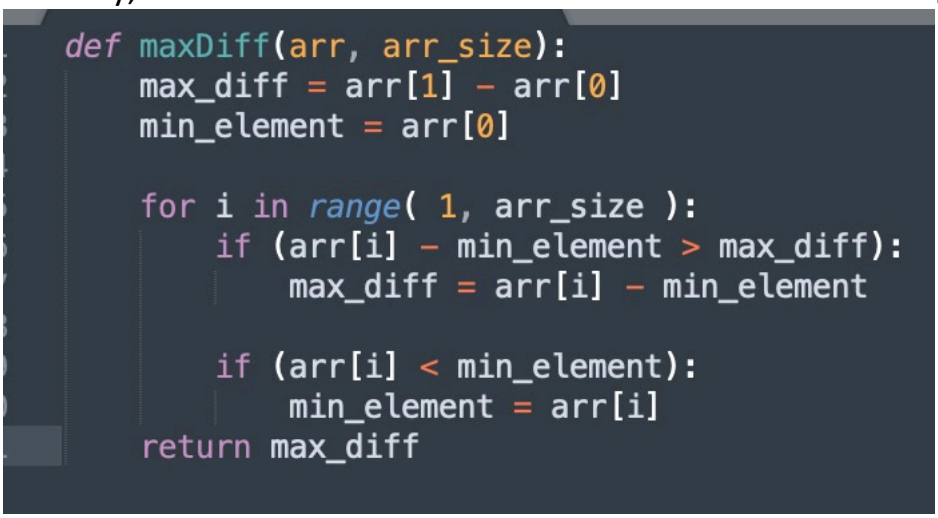
5. How to calculate the n'th Fibonacci number.

```python
class Solution:
    def fib(self, N: int) -> int:
        lastlast, last = 0,1
        for _ in range(N):
            lastlast, last = last, last+lastlast
        return lastlast
```

6. Technical Question based on finding the largest difference between two positive integer lists

7. Create a function to parse a XML file

8. Sorted array, find one element with the least difference to the target



```
def maxDiff(arr, arr_size):
    max_diff = arr[1] - arr[0]
    min_element = arr[0]

    for i in range( 1, arr_size ):
        if (arr[i] - min_element > max_diff):
            max_diff = arr[i] - min_element

        if (arr[i] < min_element):
            min_element = arr[i]
    return max_diff
```

9. Return the nth to last element in a linked list and and array rotation question.

```python
class Solution(object):
    def removeNthFromEnd(self, head, n):
        """
        :type head: ListNode
        :type n: int
        :rtype: ListNode
        """
        prev = ListNode(0)
        prev.next = head
        del_node = prev

        for i in range(n):
            del_node = del_node.next

        while(del_node.next):
            del_node = del_node.next
            prev = prev.next

        if prev.next == head:
            return head.next
        else:
            prev.next = prev.next.next
            return head
```

10. Given an array and a target number, find an element in the array that is the closest to the target num.

11. Determine if a linked list is a palindrome

```python
class Solution:
    def isPalindrome(self, head):
        """
        :type head: ListNode
        :rtype: bool
        """
        if not head or not head.next:
            return True
        slow, fast = head, head
        while(fast.next and fast.next.next):
            slow = slow.next
            fast = fast.next.next
        # find the latter half of the linkedlist
        slow = slow.next
        slow = self.reverseLinkedList2(slow)

        while(slow):
            if head.val != slow.val:
                return False
            head = head.next
            slow = slow.next
        return True

    def reverseLinkedList(self, head):
        new_head = None
        while(head):
            p = head
            head = head.next
            p.next = new_head
            new_head = p
        return new_head

    def reverseLinkedList2(self, head):
        if not head or not head.next:
            return head
        p = self.reverseLinkedList2(head.next)
        head.next.next = head
        head.next = None
        return p
```

12. Given an array of integers, checking if every element has a pair.

13. Convert string representing an integer to an integer without using built in functions

```
class Solution:
    def myAtoi(self, str: str) -> int:
        res = 0
        n = len(str)
        if not n:   return res

        i, sign = 0, 1
        while(i<n):
            if str[i] == '-':
                sign = -1
                i +=1
                break
            elif str[i] == '+':
                i +=1
                break
            elif str[i].isdigit():
                break
            elif str[i] == ' ':
                i+=1
            else:
                return res
        while(i<n and str[i].isdigit()):
            res = res*10 + int(str[i])
            i += 1
        res *= sign
        if res > pow(2,31)-1:
            return pow(2,31)-1
        elif res < -pow(2,31):
            return -pow(2,31)
        else:
            return res
```

14. Give the minimum height of a binary search tree?

15. https://www.geeksforgeeks.org/relationship-number-nodes-height-binary-tree/

16.Search for an element in a sorted but rotated array

```python
class Solution:
    def search(self, nums, target):
        """
        :type nums: List[int]
        :type target: int
        :rtype: int
        """
        # 二分法的关键在于取中间值后，与目标值比较，判断左半段还是右半段。
        # 观察上述八种情况，可以发现以中间值为界，左右两部分总有一段是绝对升序的。
        # 当 nums[mid] < nums[right] 时，右半段绝对升序；
        # 当 nums[mid] > nums[right] 时，左半段绝对升序
        n = len(nums)
        if n==0:
            return -1
        l,r = 0, n-1
        while(l<r):
            mid = (l+r) // 2
            print(l,r,mid)
            if nums[mid] == target:
                return mid
            if nums[mid] < nums[r]:
                if nums[mid] < target and nums[r] >= target:
                    l = mid + 1
                else:
                    r = mid -1
            elif nums[mid] > nums[r]:
                if nums[l] <= target and nums[mid] > target:
                    r = mid -1
                else:
                    l = mid +1
        if l == r and nums[l] == target:
            return l
        else:
            return -1

```

Given two unsorted lists of positive integer numbers, find the largest difference between the two. The number in list1 must be smaller than list2.

Example:
{1, 3, 8, 7, 4}
{0, 2, 6, 5, 10}

Largest difference is between 1 and 10.

```
def findSmallestDifference(A, B):

    # Sort both arrays
    # using sort function
    m = len(A)
    n = len(B)
    A.sort()
    B.sort()
    i = 0
    j = 0

    # Initialize result as max value
    result = float('inf')

    # Scan Both Arrays upto
    # sizeof of the Arrays
    while (i < m and j < n):

        if (abs(A[i] - B[j]) < result):
            result = abs(A[i] - B[j])

        # Move Smaller Value
        if (A[i] < B[j]):
            i += 1
        else:
            j += 1
    # return final sma result
    return result
```

Given two integer arrays sorted in ascending order and an integer k. Define sum = a + b, where a is an element from the first array and b is an element from the second one. Find the kth smallest sum out of all possible sums.

Give the minimum height of a binary search tree?