1291. Sequential Digits

An integer has *sequential digits* if and only if each digit in the number is one more than the previous digit.

Return a **sorted** list of all the integers in the range `[low, high]` inclusive that have sequential digits.

Example 1:

```
Input: low = 100, high = 300
Output: [123,234]
```

Example 2:

```
Input: low = 1000, high = 13000
Output: [1234,2345,3456,4567,5678,6789,12345]
```

Constraints:

- `10 <= low <= high <= 10^9`

```
def sequentialDigits(self, low: int, high: int) -> List[int]:
        res = []
        firstdigit = int(str(low)[0])
        for i in range(1,10):
            self.permute(res,low,high,str(i))
        return sorted(res)

    def permute(self,res,low,high,ssf):
        if low<=int(ssf)<=high:
            res.append(int(ssf))
            # return

        for i in range(int(ssf[-1]),10):
            if i-int(ssf[-1])==1:
                self.permute(res,low,high,ssf+str(i))
```

```
class Solution:
    def sequentialDigits(self, low, high):
        out = []
        queue = deque(range(1,10))
        while queue:
            elem = queue.popleft()
```

```
            if low <= elem <= high:
                out.append(elem)
            last = elem % 10
            if last < 9: queue.append(elem*10 + last + 1)

        return out
```