

cenkay ★ 3413 Last Edit: October 8, 2019 3:55 PM 499 VIEWS

DFS

```
class Solution:
    def countSteppingNumbers(self, low: int, high: int) -> List[int]:
        def dfs(n):
            if n > high:
                return
            if n >= low:
                q.add(n)
            d = n % 10
            if d == 0:
                dfs(n * 10 + 1)
            elif d == 9:
                dfs(n * 10 + 8)
            else:
                dfs(n * 10 + d + 1)
                dfs(n * 10 + d - 1)
        q = set()
        for i in range(10):
            dfs(i)
        return sorted(q)
```

Queue

```
class Solution:
    def countSteppingNumbers(self, low: int, high: int) -> List[int]:
        q1, q2 = set(), collections.deque(range(10))
        while q2:
            n = q2.popleft()
            if low <= n <= high:
                q1.add(n)
            if n < high:
                d = n % 10
                if d == 0:
                    q2.append(n * 10 + 1)
                elif d == 9:
                    q2.append(n * 10 + 8)
                else:
                    q2.append(n * 10 + d + 1)
                    q2.append(n * 10 + d - 1)
        return sorted(q1)
```

Comments: 4 Best Most Votes Newest to Oldest Oldest to Newest

Type comment here... (Markdown is supported)

Post

DennyZhang ★ 225 October 8, 2019 11:33 AM

Try not sorting it at the end. It will slow down the algorithm by a lot.

0 🗨️ Reply

TheAmazingLizard ★ 150 October 7, 2019 8:23 PM

Great solutions, thank you!

Can the DFS solution be improved by checking if `n >= low` before adding it to the queue, that way you don't have to filter at the end?

0 🗨️ Show 1 reply 🗨️ Reply

sa7810 ★ 91 Last Edit: October 6, 2019 4:59 AM

Can you also comment on the time complexity of your solution? Thanks.
My guess is it is 2^{10} . Is that correct?

0 🗨️ Show 1 reply 🗨️ Reply

routesf ★ 9 October 6, 2019 3:51 AM

great solution, thanks for sharing.

0 🗨️ Reply