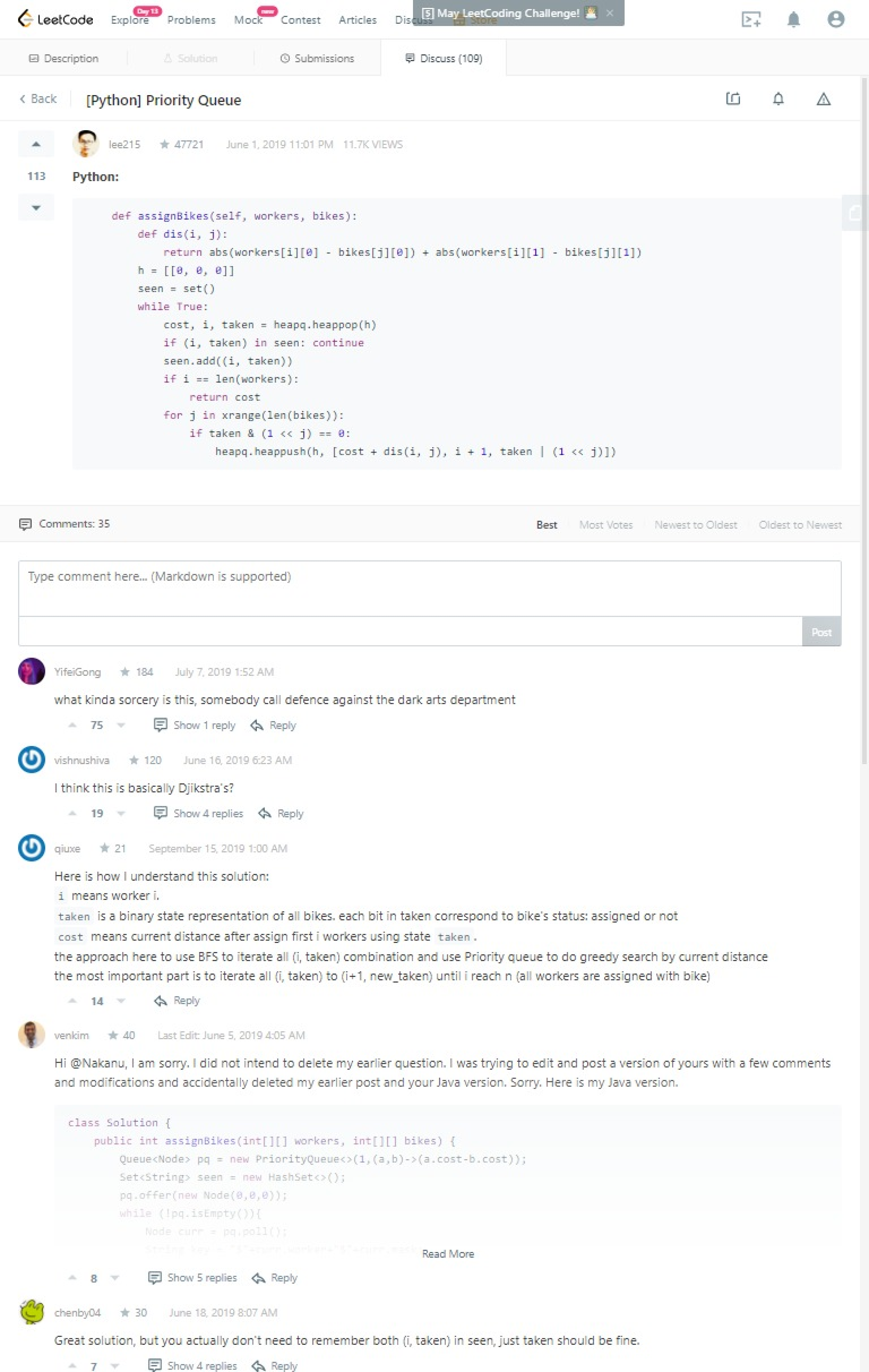

[Python] Priority Queue

lee215 ★ 47721 June 1, 2019 11:01 PM 11.7K VIEWS

113

Python:

```
    def assignBikes(self, workers, bikes):
        def dis(i, j):
            return abs(workers[i][0] - bikes[j][0]) + abs(workers[i][1] - bikes[j][1])
        h = [[0, 0, 0]]
        seen = set()
        while True:
            cost, i, taken = heapq.heappop(h)
            if (i, taken) in seen: continue
            seen.add((i, taken))
            if i == len(workers):
                return cost
            for j in xrange(len(bikes)):
                if taken & (1 << j) == 0:
                    heapq.heappush(h, [cost + dis(i, j), i + 1, taken | (1 << j)])
```

Comments: 35

YifeiGong ★ 184 July 7, 2019 1:52 AM

what kinda sorcery is this, somebody call defence against the dark arts department

75 Show 1 reply Reply

vishnushiva ★ 120 June 16, 2019 6:23 AM

I think this is basically Djikstra's?

19 Show 4 replies Reply

qiuxe ★ 21 September 15, 2019 1:00 AM

Here is how I understand this solution:
`i` means worker i.
`taken` is a binary state representation of all bikes. each bit in taken correspond to bike's status: assigned or not
`cost` means current distance after assign first i workers using state `taken`.
the approach here to use BFS to iterate all (i, taken) combination and use Priority queue to do greedy search by current distance
the most important part is to iterate all (i, taken) to (i+1, new_taken) until i reach n (all workers are assigned with bike)

14 Reply

venkim ★ 40 Last Edit: June 5, 2019 4:05 AM

Hi @Nakanu, I am sorry. I did not intend to delete my earlier question. I was trying to edit and post a version of yours with a few comments and modifications and accidentally deleted my earlier post and your Java version. Sorry. Here is my Java version.

```
class Solution {
    public int assignBikes(int[][] workers, int[][] bikes) {
        Queue<Node> pq = new PriorityQueue<>(1,(a,b)->(a.cost-b.cost));
        Set<String> seen = new HashSet<>();
        pq.offer(new Node(0,0,0));
        while (!pq.isEmpty()){
            Node curr = pq.poll();
            String key = "$"+curr.worker+"$"+curr.mask
```

Read More

8 Show 5 replies Reply

chenby04 ★ 30 June 18, 2019 8:07 AM

Great solution, but you actually don't need to remember both (i, taken) in seen, just taken should be fine.

7 Show 4 replies Reply