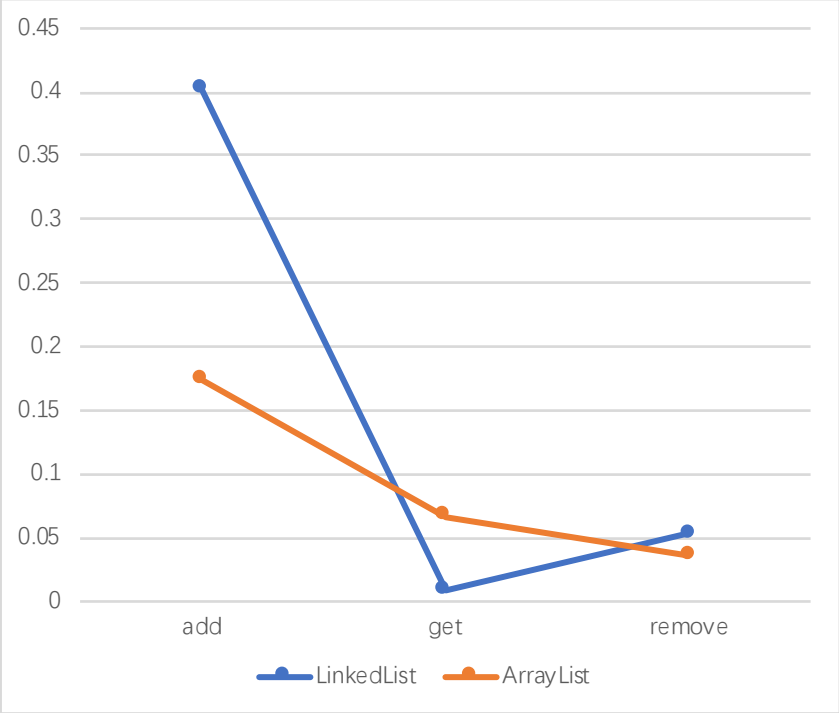
Result:

```
chengaoxiangdeMacBook-Pro:HW3 chengaoxiang$ java HW3
ArrayList add: 0.174322606s
LinkedList add: 0.402240048s
ArrayList get(Iterator): 0.067197278s
ArrayList get: 0.047370396s
LinkedList get(Iterator): 0.008677800s
LinkedList get: 1617.206622422s
ArrayList remove(front to end): 188.490113088
ArrayList remove(end to front): 0.036161386s
LinkedList remove(front to end): 0.041012206
LinkedList remove(end to frond))  : 0.053243621s
```



As what I think, when the spend of those operations depend on the direction.
For arraylist, when add/remove element from front to end, it will move all the array, so it cost tons of time, O(n). While linkedlist is faster, O(1).
For linkedlist, when get element without using iterator, since it will search element from front to end, it will cost tons of time, O(n^2). While get function in arraylist is much faster, O(1).
When using iterator, linkedlist get is faster than arraylist, when from end to frond, arraylist remove is faster than linkedlist.