

HOUSE ROBBER 2

Similar to House Robber 1 only difference is that the houses are all connected and our circular which indicates the first index & the last index are also adjacent elements.

Sol

We can solve the answer with the same logic as house robber 1 with just a small update which is:

- a) We'll leave out the last element & apply the same house robber 1 to all the remaining elements (or)
- b) We can leave out the first element & apply the same house robber 1 logic to all the remaining elements

If let's say a) is ans₁ & b) is ans₂
(leaving last) (leaving first)

& can just return the answer to this question is
 $\max(\text{ans}_1, \text{ans}_2)$

#

Code

```
HouseRobber1(List<Integer> nums) {
    int n = nums.size();
    if (n == 0) return 0;
    if (n == 1) return nums.get(0);
    int prev = nums.get(0);
    int prev2 = 0;
    for (int i = 1; i < n; i++) {
        int take = nums.get(i);
        if (i > 1) take += prev2;
        int notTake = 0 + prev;
        int curr = Math.max(take, notTake);
        prev2 = prev;
        prev = curr;
    }
    return prev;
}

houseRobber2(int[] houses) {
    int n = houses.length;
    if (n == 1) return houses[0];
    List<Integer> temp1 = new ArrayList<>();
    List<Integer> temp2 = new ArrayList<>();
    for (int i = 0; i < n; i++) {
        if (i != 0) temp1.add(houses[i]);
        if (i != n - 1) temp2.add(houses[i]);
    }
    return Math.max(
        max HouseRobber1(temp1),
        HouseRobber1(temp2));
}
```