

(2) House Robbery : Given houses with money, without robbing any adjacent houses, rob max. money:

```
int robMax(int [] house) {  
    if (house.length == 0) return 0;  
    int house_a = house[0];  
    if (house.length == 1) return house[0], house_a;  
    int house_b = house[1];  
    if (house.length == 2)  
        return Math.max(house[0], house[1]);  
    return Math.max(house[0], house[1]);
```

```
int [] robbery; ++way; ++qot; -100;  
int houseLastRobbed = house[0];  
int maxRobbed = Math.max(house_a, house_b);
```

```
int maxRobbed = lastRobbed;
```

```
for (int i=2; i < house.length; i++) {
```

O(n) // Pick max of previous house or (this house + lastRobbed)  
maxRobbed = Math.max(house[i] + lastRobbed, house[i]);  
house\_a = house\_b;  
house\_b = maxRobbed;

```
}
```

```
return maxRobbed;
```