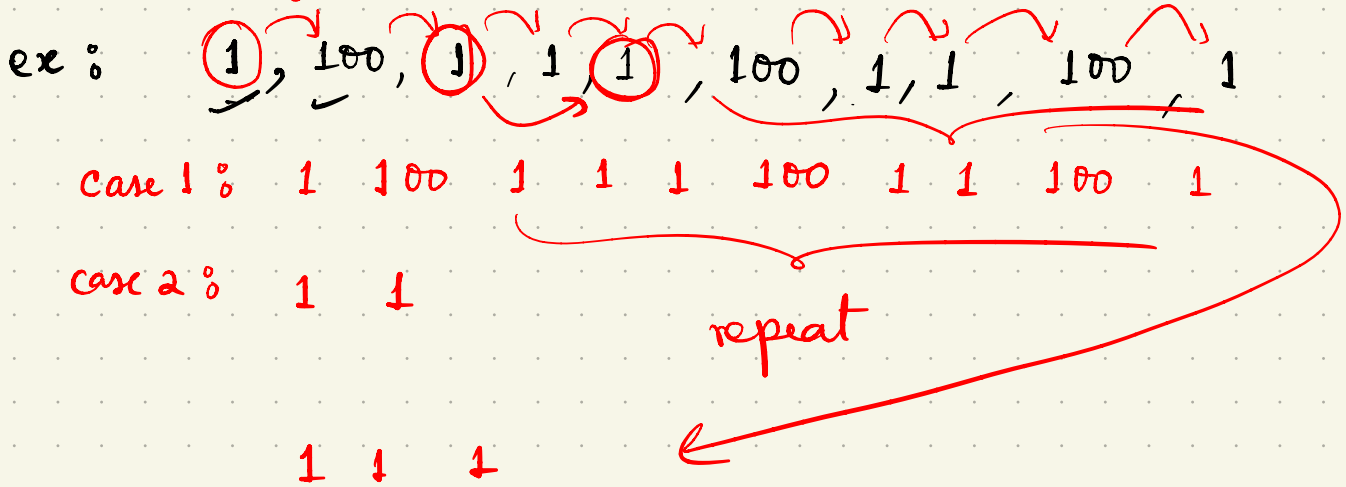
Min cost climbing stairs :

The question state that we've given an array of cost and we've to reach the end or out of bound.

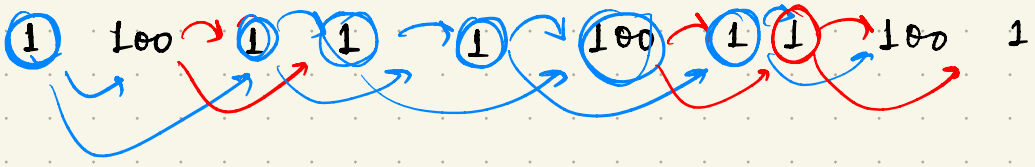
And it's also mention that we can jump 1 or 2 step at time and we can start from 0 or

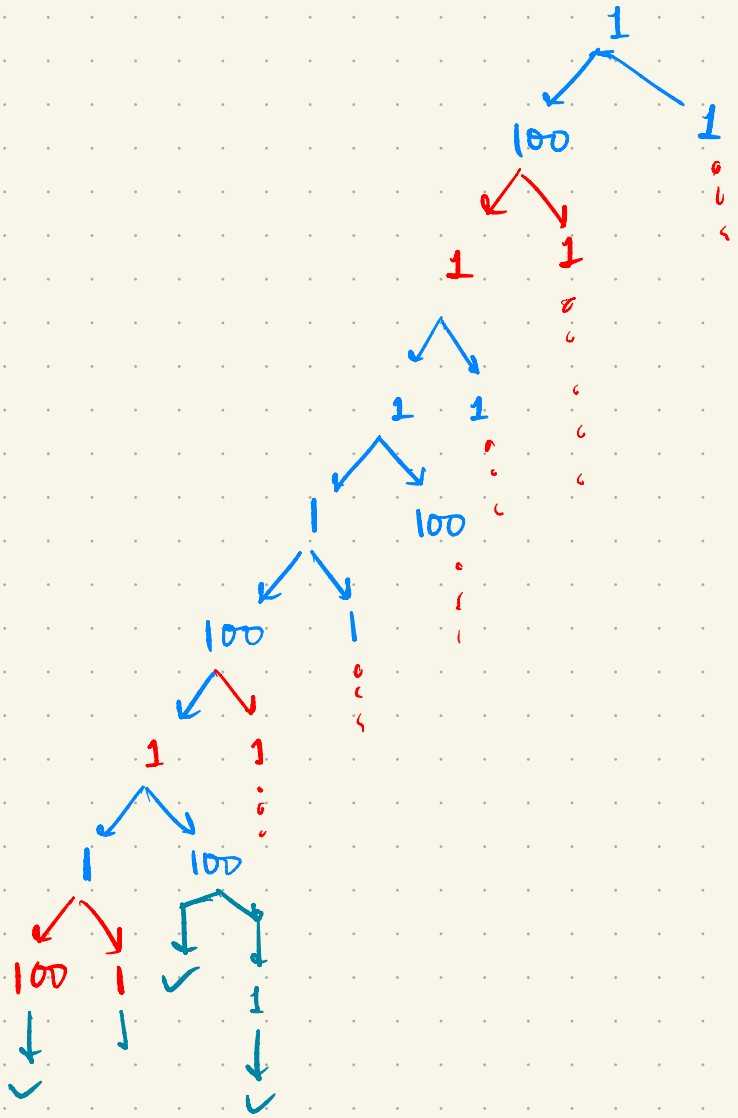
* The first thing here is to look that we've to find the minimum cost and while finding it will be necessary to check both the element at index+1 or index+2. because there might be possiblity we can get a minimum solution.



1 1 1

0
6
0
0
.
.
6





As we can see the TC is exponential so it'll give as a answer but it'll very time consuming.

m^m (possiblity) exponential time complexity

In the above mention total possiblities there will be one required possiblity that it the answer. But how we can find it?

Whenever the question is saying find the min and and max sum result and we've explore all the cases to find the desired outcome the way to do it's

RECURSION.

Let's write the recursive code:

```
int fun( arr,n) {
    if( n == 0 || n == 1) return arr[n];
    return ( arr [n] + min( fun( arr, n-1), fun(arr, n-2))
}
```

How we can optime the code? → From observation there are cases which are repeating many time. If we can eliminate them may be then our code will be more optimise. How we can do that?

DP

```
int fun(arr,n){
    if(n==0 || n==1) return arr[n];
    if(dp[n]!=-1) return dp[n];

    dp[n]=(arr[n] + min(fun(arr,n-1),fun(arr,n-2))

    return dp[n];
}
```

If we can write the recursive code then converting it do the dp code won't be much difficult.