```
def init points:
    find low_init, high_init within data[0:depth]
    points.append(low_init, high_init).

def init position:
    if len(points) < 2     init points, return points[1]
    if len(points) ≥ 2     return points[-2]

def find points: <power=1
    tmp_high, tmp_low = max(data[previous.idx : +power*depth]), min(data[previous.idx : +power*depth])
    if previous = low point:
        if tmp_high > previous + dev% ,
            points.append(tmp_high)
            previous = tmp_high.
        elif tmp_low < previous
            points[-1] = tmp_low
            previous = tmp_low. <else: power += 1.
    if previous = high point:
        if tmp_high > previous
            points[-1] = tmp_high
            previous = tmp_high.
        elif tmp_low < previous - dev%.
            points.append(tmp_low)
            previous = tmp_low. <else: power += 1
    if previous.idx + depth ≥ len(data)
        end
```

process the one according to idx