Bézier Curves using De Casteljau's algorithm

```
function lerp(t,xList,YList){
    order = XList.length-1;
    var xOutList = new Array(order).fill(0);
    var yOutList = new Array(order).fill(0);
    for (let i=0; i<order;i++){
        xOutList[i] = t*xList[i]+(1-t)*xList[i+1];
        yOutList[i] = t*yList[i]+(1-t)*yList[i+1];
    }
    return [xOutList,yOutList];
}
```

Core of De Casteljau Algorithm

- Input: Random Control Points $\mathcal{P}$: $\{P_0^{(0)}, \cdots, P_n^{(0)}\}$, time t
- Output : $P(t)$

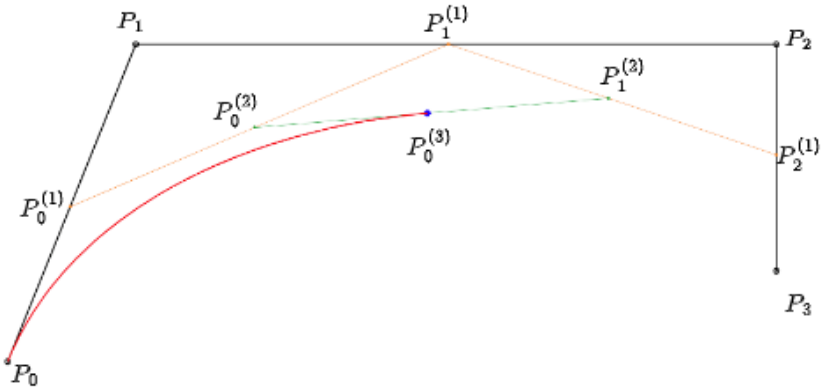
1. Compute $\mathcal{P}^{(1)} = \{P_0^{(1)}, \cdots, P_{n-1}^{(1)}\}$ Using *lerp* function;

$$P_i^{(1)} = t * P_i^{(0)} + (1-t) * P_{i+1}^{(0)}$$

2. Compute $\mathcal{P}^{(2)} = \{P_0^{(2)}, \cdots, P_{n-2}^{(2)}\}$ using *lerp* function;

$$P_i^{(r)} = t * P_i^{(r-1)} + (1-t) * P_{i+1}^{(t-1)}$$

3. $\cdots$
4. Compute $\mathcal{P}^{(n)} = \{P_0^n\}$


P_1
$P_1^{(1)}$
P_2
$P_1^{(2)}$
$P_0^{(2)}$
$P_0^{(3)}$
$P_2^{(1)}$
$P_0^{(1)}$
P_3
P_0