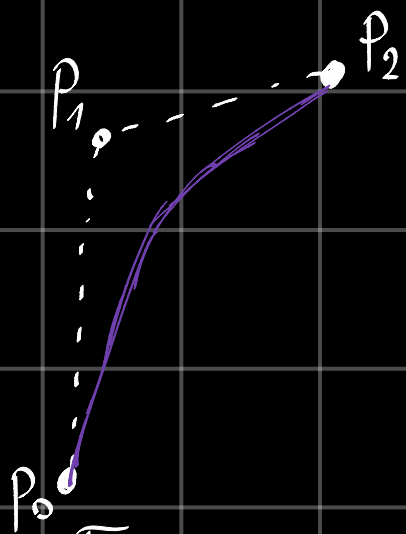


# Krzywe Bézierego stopnia $n$

Mamy punkty  $p_0, \dots, p_m$  i tworzą funkcje parametryczne  $p(t) = \sum_{i=0}^n p_i B_i^n(t)$  dla  $t \in [0, 1]$   
 $B_i^n(t) = \binom{n}{i} t^i (1-t)^{n-i}$  wielomiany Bernstein



Formuła rekurencyjna

$$p_i^0 = p_i \quad i = 0, \dots, n$$

$$p_i^j = p_i^{j-1}(1-t_0) + p_{i+1}^{j-1}t_0 \quad i = 0, \dots, n \quad j = 1, \dots, n$$

## Podjęcie:

Wrzucamy punkty na GPU,  
odpalamy compute shadera  
do wyliczenia punktów przez  
które przechodzi funkcja parametryczna  
dla dokładności liczymy tylko to co  
wiolszy z dokładnością do pikselu  
i na koniec fragment shader do rysowania  
po naszych punktach

## Optymalizacje:

- 1) renderujemy tylko to co na ekranie
- 2) Aktualizacja tylko po edycji punktów lub przesunięciu, przybliżeniu (przybliżenie, przesunięcie)
- 3) Aktualizujemy to co musimy.