

```
let rec roots ~eps poly =  
  match Polynomial.degree poly with  
  | non_nat when non_nat <= 0 -> []  
  | 1 -> poly |> linear_root |> Option.to_list  
  | 2 -> poly |> quadratic_roots ~eps  
  | _ ->  
    poly  
    |> Polynomial.derivative  
    |> roots ~eps  
    |> I.intervals_of_list  
    |> List.filter_map ~f:(BS.search ~f:(fun x -> Polynomial.calc poly ~x) ~eps)  
;;
```