

Programación: Interpolación segmentaria lineal

1. Búsqueda del segmento (2 %). Escriba una función `FindSegment` de los argumentos `xs`, `x` que regrese el índice `k` tal que $xs[k] \leq x < xs[k + 1]$. Se supone que

$$xs[1] < xs[2] < \dots < xs[n], \quad xs[1] \leq x \leq xs[n],$$

donde `n` es la longitud de la lista `xs`. Para $x = xs[n]$ la función tiene que regresar `n - 1`.

2. Comprobación.

`FindSegment[{10,20}, 10]` la respuesta correcta es 1;

`FindSegment[{10,12,15,19,20,29,30}, 14]` la respuesta correcta es 2;

`FindSegment[{10,12,15,19,20,29,30}, 29]` la respuesta correcta es 6;

`FindSegment[{10,12,15,19,20,29,30}, 30]` la respuesta correcta es 6.

3. Evaluación de una función lineal a trozos (2 %). Escriba una función `LinSpline` de los argumentos `xs`, `as`, `bs`, `x` que calcule la función lineal a trozos definida por la regla:

$$S(x) = as[k] + bs[k] * (x - xs[k]), \quad \text{si} \quad xs[k] \leq x < xs[k + 1].$$

Para $x = xs[n]$ la función tiene que aplicar la misma fórmula que en el intervalo $[xs[n - 1], xs[n])$ y regresar

$$as[n - 1] + bs[n - 1] * (xs[n] - xs[n - 1]),$$

Indicación: la función `LinSpline` puede llamar la función `FindSegment`.

4. Prueba.

```
xs = {1, 5, 7, 8}; as = {3, -1, 0}; bs = {-1, 1/2, 3};
```

```
Plot[linspline[xs, as, bs, x], {x, -1, 8}]
```

5. Cálculo de los coeficientes de la interpolación segmentaria lineal (2 %). Escriba una función `CoefsLinSpline` de los argumentos `xs`, `ys` que construya y regrese las listas `as`, `bs` de los coeficientes de la interpolación segmentaria lineal.

6. Comprobación.

`CoefsLinSpline[{1,5,7,8},{3,-1,0,3}]` debe regresar `{{3,-1,0},{-1,1/2,3}}`.

7. Prueba total.

```
xs = {0,1.5,3,4,5,7}; ys = Map[Sin, xs];
```

```
{as, bs} = CoefsLinSpline[xs, ys];
```

```
Plot[{Sin[x], LinSpline[xs, as, bs, x]}, {x, 0, 7}];
```