Programación:
Construcción del polinomio interpolante con la fórmula de Newton

Se recomienda resolver estos ejercicios **antes** de la clase de programación.

Objetivos. Programar la construcción del polinomio interpolante con la fórmula de Newton, usando la tabla de diferencias divididas.

Requisitos. Cálculo de los valores del polinomio en una lista de puntos, multiplicación de un polinomio por un binonio.

1. Funciones que trabajan con polinomios. Suponemos que los polinomios se guardan como listas de coeficientes, empezando con el coeficiente de x^0. Vamos a usar las funciones PolValues y MulPolBinom de las clases anteriores:

> `PolValues` de dos argumentos `fcoefs`, `points` calcula los valores del polinomio con coeficientes `fcoefs` en los puntos de la lista `points`;
>
> `MulPolBinom` de dos argumentos `fcoefs`, `b` calcula los coeficientes del producto del polinomio con coeficientes `fcoefs` por el binomio $(x + b)$. Atención: el binomio $(x + b)$ está dado no por la lista de sus coeficientes, sino por un número `b`.

Cálculo de la tabla de las diferencias divididas

2. Definición de las diferencias divididas. Supongamos que $x_1, \ldots, x_n$ son algunos números diferentes a pares y pertenecientes al dominio de definición de una función f, y denotemos por $y_1, \ldots, y_n$ a los valores de la función f en estos puntos. Entonces las *diferencias divididas* de la función f en estos puntos se definen de manera recursiva:

$$f[x_i, x_{i+1}, \ldots, x_{j-1}, x_j] := \frac{f[x_{i+1}, \ldots, x_{j-1}, x_j] - f[x_i, x_{i+1}, \ldots, x_{j-1}]}{x_j - x_i}, \qquad (1)$$

con las condiciones iniciales $f[x_i] = y_i$. Suponemos que los números $x_1, \ldots, x_n$ (diferentes a pares) y los números $y_1, \ldots, y_n$ son nuestros datos iniciales; el problema es calcular las siguientes diferencias divididas:

$$f[x_1], \qquad f[x_1, x_2], \qquad f[x_1, x_2, x_3], \qquad f[x_1, x_2, x_3, \ldots, x_n].$$

3. Escriba las fórmulas recursivas (1) para los siguientes casos particulares ($n = 3$):

$f[x_1] = y_1$

$f[x_2] = y_2 \qquad f[x_1, x_2] =$ ————————

$f[x_3] = y_3 \qquad f[x_1, x_2] =$ ———————— $\qquad f[x_1, x_2, x_3] =$ ————————

4. Idea recomendada: guardar las diferencias divididas en un arreglo unidimensional borrando los valores que ya no se necesitan. Dados los puntos $x_1, \ldots, x_n$ y los valores $y_1, \ldots, y_n$ tenemos que calcular sus diferencias divididas. Vamos a guardarlas en un arreglo `dd` de longitud n, como se muestra en el siguiente esquema para $n = 4$. Están escritos con color gris los contenidos que ya no se cambian, sino se heredan de los pasos anteriores.

	Paso 0	Paso 1	Paso 2	Paso 3
contenido de `dd[1]`:	$f[x_1]$	$f[x_1]$	$f[x_1]$	$f[x_1]$
contenido de `dd[2]`:	$f[x_2]$	$f[x_1, x_2]$	$f[x_1, x_2]$	$f[x_1, x_2]$
contenido de `dd[3]`:	$f[x_3]$	$f[x_2, x_3]$	$f[x_1, x_2, x_3]$	$f[x_1, x_2, x_3]$
contenido de `dd[4]`:	$f[x_4]$	$f[x_3, x_4]$	$f[x_2, x_3, x_4]$	$f[x_1, x_2, x_3, x_4]$

En el p-ésimo paso se calculan las diferencias divididas del p-ésimo orden. En casa paso (es decir, en cada columna) tenemos que empezar desde las últimas entradas. Por ejemplo, sería un error empezar en el Paso 1 con

$$\texttt{dd}[2] \coloneqq \frac{\texttt{dd}[2] - \texttt{dd}[1]}{x[2] - x[1]}$$

porque de esta manera se perdería el valor anterior $\texttt{dd}[2] = f[x_2]$, el cual todavía necesitamos para calcular $f[x_2, x_3]$. Mencione en el orden correcto los valores que calculamos en cada paso.

Paso 0 (inicialización): `dd` es una copia del arreglo $y = [y_1, \ldots, y_n]$.

Paso 1: $\texttt{dd}[4] \coloneqq f[x_3, x_4]$, $\texttt{dd}[\quad] \coloneqq f[\quad]$, $\texttt{dd}[\quad] \coloneqq f[\quad]$.

Paso 2: $\texttt{dd}[4] \coloneqq f[x_2, x_3, x_4]$, $\texttt{dd}[\quad] \coloneqq f[\quad]$.

Paso 3: $\texttt{dd}[4] \coloneqq f[\quad]$.

5. Tabla de las diferencias divididas, seguimiento por pasos, $n = 4$.

Paso $p = 0$.

$dd \coloneqq$ una copia del arreglo $y[1], y[2], y[3], y[4]$.

Paso $p = 1$.

$dd[4] \coloneqq (dd[4] - dd[3]) \;/\; (x[4] - x[3])$

$dd[3] \coloneqq (dd[3] - dd[2]) \;/\; (x[3] - x[2])$

$dd[\;\;] \coloneqq (dd[\;\;] - dd[\;\;]) \;/\; (x[\;\;] - x[\;\;])$

Paso $p = 2$.

$dd[\;\;] \coloneqq (dd[\;\;] - dd[\;\;]) \;/\; (x[\;\;] - x[\;\;])$

$dd[3] \coloneqq (dd[3] - dd[2]) \;/\; (x[3] - x[1])$

Paso $p = 3$.

$dd[\;\;] \coloneqq (dd[\;\;] - dd[\;\;]) \;/\; (x[\;\;] - x[\;\;])$

6. Tabla de las diferencias divididas, estructura de los ciclos, $n = 4$.

$dd \coloneqq$ una copia del arreglo y.

Paso $p = 1$. Para $j \coloneqq \;\;,\;\;,\;\;:$

$dd[j] \coloneqq$ ————————;

Paso $p = 2$. Para $j \coloneqq \;\;,\;\;:$

$dd[j] \coloneqq$ ————————;

Paso $p = 3$. Para $j \coloneqq \;\;:$

$dd[j] \coloneqq$ ————————;

Regresar dd.

7. Tabla de las diferencias divididas. Escriba una función `DivDifTable` de dos argumentos `points` y `values` que para los puntos `points` $= (x_1, \ldots, x_n)$ y los valores correspondientes `values` $= (y_1, \ldots, y_n)$ calcule y regrese la lista de las siguientes diferencias divididas:

$$[y_1], \qquad [y_1, y_2], \qquad [y_1, y_2, y_3], \qquad \ldots, \qquad [y_1, y_2, y_3, \ldots, y_n].$$

```
Entrada: points, values.
n := longitud de la lista points;
dd := copia de la lista values;
Para p := ???, ..., ???:
    Para j := ???, ..., ???:
        dd[j] := (dd[???] - dd[???]) / (points[???] - points[???]);
Regresar dd.
```

8. Traduzca el pseudocódigo escrito en el ejercicio anterior a algún lenguaje de programación en el cual va a realizarlo.

9. Prueba de la función DivDifTable. Aplique la función a los puntos $2, 3, -1, 5$ y los valores $3, 11, -9, 69$. La respuesta correcta es la lista $3, 8, 1, 1$.

10. Número de operaciones. Calcule el número de operaciones de división en el algoritmo DivDifTable. Por tener dos ciclos encajados, obtendremos un polinomio de segundo grado:

$$\sum_{p=???}^{???} \sum_{j=???}^{???} ??? = \qquad\qquad = \underbrace{\qquad}_{?} n^2 + \underbrace{\qquad}_{?} n^1 + \underbrace{\qquad}_{?} n^0.$$

Construcción del polinomio interpolante

11. Fórmula de Newton para el polinomio interpolante.
Para los puntos $x_1, \ldots, x_n$ y los valores $y_1, \ldots, y_n$, el polinomio interpolante se puede construir mediante la siguiente fórmula:

$$P(x) = \sum_{k=1}^{n} f[x_1, \ldots, x_k] \prod_{j=1}^{k-1} (x - x_j). \tag{2}$$

Por ejemplo, para $n = 3$,

$$P(x) = f[x_1] + f[x_1, x_2](x - x_1) + f[x_1, x_2, x_3](x - x_1)(x - x_2).$$

12. Ejemplo ($n = 4$). Escriba la fórmula (2) para $n = 4$:

$$P(x) =$$

13. Truco de Horner. La expresión (2) se puede calcular de manera eficiente, usando el siguiente truco que se encuentra en trabajos de varios matemáticos, incluso Qin Jiushao, Paolo Ruffini y William George Horner. Para cualesquiera números $a, b, c, u_1, u_2, u_3, u_4$,

$$u_1 + u_2 a + u_3 ab + u_4 abc = (((u_4)c + u_3)b + u_2)a + u_1.$$

Haga la misma transformación para $n = 5$:

$$u_1 + u_2 a + u_3 ab + u_4 abc + u_5 abcd =$$

14. Representación eficiente de la fórmula de Newton para $n = 4$.

$$P(x) = \Big(\Big(f[x_1, x_2, x_3, x_4](x - x_3) + f[x_1, x_2, x_3] \Big)(x - x_2) + f[x_1, x_2] \Big)(x - x_1) + f[x_1].$$

Escriba los cálculos por pasos. Invente una numeración conveniente de los pasos.

$P(x) := f[x_1, x_2, x_3, x_4];$

Paso ???:
$P(x) := P(x) \cdot (x - x_3);$ $\quad P(x) := P(x) + f[x_1, x_2, x_3];$

Paso ???:
$P(x) := P(x) \cdot (x -$ $);$ $\quad P(x) := P(x) +$ $;$

Paso ???:
$P(x) := P(x) \cdot (x -$ $);$ $\quad P(x) := P(x) +$ $.$

15. ¿Cómo realizar la multiplicación de un polinomio por un binomio? Dada la lista `pcoefs` de los coeficientes de un polinomio $P(x)$ y un número `a`, necesitamos calcular los coeficientes del producto $P(x)(x-a)$. Recuerde cuál función nos puede servir. ¿Con qué argumentos tenemos que llamarla? Escriba la respuesta:

16. ¿Cómo realizar la adición de una constante a un polinomio? Dada la lista `pcoefs` de los coeficientes de un polinomio $P(x)$ y un número `c`, necesitamos calcular los coeficientes de la suma $P(x)+c$. ¿Cuál coeficiente de la lista `pcoefs` tenemos que cambiar y de qué manera? Escriba la respuesta:

17. Construcción del polinomio interpolante con la fórmula de Newton. Esciba una función `NewtonInterpol` con argumentos `points` y `values` que construya el polinomio interpolante correspondiente a los puntos `points` y valores `values`. Use las funciones `DivDifTable` y `MulPolBinom`. Esquema del algoritmo:

```
Entrada: points, values;
dd := DivDifTable(points, values);
pcoefs := lista de un elemento ???;
Para k := ???, ..., ???:
    pcoefs := MulPolBinom(pcoefs, ???);
    pcoefs[???] += ???;
Regresar pcoefs.
```

18. Prueba. Usando la función `NewtonInterpol` calcule los coeficientes del polinomio interpolante para los puntos $2, 3, -1, 5$ y los valores $3, 11, -9, 69$. Haga la comprobación usando la función `PolValues`.

19. Número de operaciones. Calcule el número de operaciones de multiplicación y división en el algoritmo NewtonInterpol. Hay que tomar en cuenta el número de operaciones de división que se realizan en DivDifTable y contar de manera correcta todas las multiplicaciones que se realizan en MulPolBinom. La respuesta es un polinomio de la variable n. ¿De qué grado es este polinomio?