1 Suppose $f(x)$ is a polynomial of degree n. Let $x_0, x_1, \ldots, x_m$ be distinct nodes. Prove that whenever $m \geq n$, the Lagrange interpolating polynomial $P_m(x)$ generated by $\{(x_i, f(x_i))\}_{i=0}^m$ is actually $f(x)$ itself.

Solution Notice that each x_i is a root of the polynomial $P_m(x) - f(x)$, which means it is an m-th order polynomial (since $m \geq n$) with $m+1$ roots. Thus, it must be identically 0, since a non-constant polynomial of degree m can have at most m real zeroes, so $P_m(x) = f(x)$ for all x.

---

2 Define $f(x) = 2^x$. Let $x_0 = -2$, $x_1 = -1$, $x_2 = 0$, $x_3 = 1$, and $x_4 = 2$.

a. Find out the interpolating polynomials $P_{0,1,4}(x)$ and $P_{1,2,3,4}(x)$.

b. Derive error bounds for them at $x = 1/2$ with respect to $f(x)$.

c. If you are allowed to only use at most three of these nodes to construct an interpolating polynomial to approximate $f(1/2)$, which three (or fewer) nodes would you choose? Why?

Solution a. The polynomials are given by

$$
\begin{aligned}
P_{0,1,4}(x) &= \frac{1}{4} \cdot \frac{(x+1)(x-2)}{-1 \cdot -4} + \frac{1}{2} \cdot \frac{(x+2)(x-2)}{1 \cdot -3} + 4 \cdot \frac{(x+2)(x+1)}{4 \cdot 3} = \frac{11}{48}x^2 + \frac{15}{16}x + \frac{29}{24} \\
P_{1,2,3,4}(x) &= \frac{1}{2} \cdot \frac{x(x-1)(x-2)}{-1 \cdot -2 \cdot -3} + 1 \cdot \frac{(x+1)(x-1)(x-2)}{1 \cdot -1 \cdot -2} + 2 \cdot \frac{(x+1)x(x-2)}{2 \cdot 1 \cdot -1} + 4 \cdot \frac{(x+1)x(x-1)}{3 \cdot 2 \cdot 1} \\
&= \frac{1}{12}x^3 + \frac{1}{4}x^2 + \frac{2}{3}x + 1
\end{aligned}
$$

b. The error bounds are

$$
\begin{aligned}
|f(1/2) - P_{0,1,4}(1/2)| &= \left| \frac{f^{(3)}(\xi)}{3!}(1/2+2)(1/2-1)(1/2-2) \right| \leq \frac{(\log 2)^3 \cdot 2^2}{3!} \cdot \frac{75}{8} = \frac{15}{4}(\log 2)^3 \\
|f(1/2) - P_{1,2,3,4}(1/2)| &= \left| \frac{f^{(4)}(\xi)}{4!}(1/2+1)(1/2)(1/2-1)(1/2-2) \right| \leq \frac{(\log 2)^4 \cdot 2^2}{4!} \cdot \frac{9}{16} = \frac{3}{32}(\log 2)^4.
\end{aligned}
$$

c. I would pick x_1, x_2, and x_3.

The two closest nodes are x_2 and x_3, which is why those are included. There's a tie between the next closest nodes, which are x_1 and x_4. I picked x_1 so that the upper bound for the derivative is minimized. Indeed, if I pick x_4, then the derivative is bounded above by $(\log 2)^3 \cdot 2^2$, whereas for x_1, the derivative is bounded above by $(\log 2)^3 \cdot 2^1$.

---

3 Approximate $f(1.6)$ using Neville's method, given that

$$f(1) = 0.75, \; f(1.3) = 0.63, \; f(1.5) = 0.55, \; f(2) = 0.49.$$

Solution For convenience, we'll overload the divided differences notation, and let $f[1, 1.3]$ be the approximation of $f(1.6)$ with the Lagrange polynomial passing through $(1, f(1))$ and $(1.3, f(1.3))$, and so on.

x	$f[x]$	
1	0.75	$f[1,1.3] = \dfrac{(1.6-1.3) \cdot 0.75 - (1.6-1) \cdot 0.63}{1-1.3} = 0.51$
1.3	0.63	$f[1.3,1.5] = \dfrac{(1.6-1.5) \cdot 0.63 - (1.6-1.3) \cdot 0.55}{1.3-1.5} = 0.51$
1.5	0.55	$f[1.5,2] = \dfrac{(1.6-2) \cdot 0.55 - (1.6-1.5) \cdot 0.49}{1.5-2} = 0.538$
2	0.49	

Next,

$$
\begin{aligned}
f[1, 1.3, 1.5] &= \frac{(1.6-1.5)f[1,1.3]-(1.6-1)f[1.3,1.5]}{1-1.5} = 0.51 \\
f[1.3, 1.5, 2] &= \frac{(1.6-2)f[1.3,1.5]-(1.6-1.3)f[1.5,2]}{1.3-2} = 0.522
\end{aligned}
$$

The last step is then given by

$$f(1.6) \approx f[1, 1.3, 1.5, 2] = \frac{(1.6-2)f[1,1.3,1.5]-(1.6-1)f[1.3,1.5,2]}{1-2} = 0.5172.$$

---

4 We apply Neville's method to approximate $f(0)$ using $f(-2)$, $f(-1)$, $f(1)$, and $f(2)$. Now suppose $f(-1)$ was mistakenly overstated by 1, while $f(2)$ was understated by 2; in other words, we are given the data

$$y_0 = f(-2),\ y_1 = f(-1)+1,\ y_2 = f(1),\ y_3 = f(2)-2,$$

to make the approximation. Determine how such mistakes affect the approximation of $f(0)$. What about the approximation of $f(x)$ for arbitrary $x \in [-2, 2]$?

Solution We proceed with Neville's method, again with the same notation as the previous problem. We'll let f' be the result from the calculation with errors.

$$
\begin{aligned}
f'[-2,-1] &= \frac{(x+2)[f(-1)+1]-(x+1)f(-2)}{-1+2} = f[-2,-1]+(x+2) \\
f'[-1,1] &= \frac{(x+1)f(1)-(x-1)[f(-1)+1]}{1+1} = f[-1,1]-\frac{1}{2}(x-1) \\
f'[1,2] &= \frac{(x-1)[f(2)-2]-(x-2)f(1)}{2-1} = f[1,2]-2(x-1) \\
f'[-2,-1,1] &= \frac{(x+2)f'[-1,1]-(x-1)f'[-2,-1]}{1+2} = f[-2,-1,1]-\frac{1}{2}(x+2)(x-1) \\
f'[-1,1,2] &= \frac{(x+1)f'[1,2]-(x-2)f'[-1,1]}{2+1} = f[-1,1,2]-\frac{2}{3}(x+1)(x-1)+\frac{1}{6}(x-2)(x-1) \\
f'[-2,-1,1,2] &= \frac{(x+2)f'[-1,1,2]-(x-2)f'[-2,-1,1]}{2+2} \\
&= f[-2,-1,1,2]-\frac{1}{6}(x+2)(x+1)(x-1)+\frac{1}{24}(x+2)(x-2)(x-1)+\frac{1}{8}(x-2)(x+2)(x-1) \\
&= f[-2,-1,1,2]-\frac{1}{6}(x+2)(x+1)(x-1)+\frac{1}{6}(x+2)(x-2)(x-1).
\end{aligned}
$$

So, these mistakes throw off the approximation of $f(x)$ by a cubic polynomial. At $x=0$, the error is

$$|f[-2,-1,1,2]-f'[-2,-1,1,2]| = \frac{1}{3}+\frac{2}{3} = 1.$$

5 Perform the procedure with $f(x) = e^x$ to approximate $f(x_*)$ with $x_* = 0.05$, $x_i = ih$, and $h = 0.1$, until the new term added can be bounded by 10^{-4}, i.e., $|P_n(x_*) - P_{n-1}(x_*)| \leq 10^{-4}$. Then derive an error bound for your approximation at x_*.

Solution I wrote code in Python to apply the divided differences method here:

```python
from math import exp

def divided_diff(nodes):
    """
    Implementation of divided differences
    Parameters:
    nodes - list containing known points on f(x)

    Output:
    List of lists containing the iterations
    """
    p = [ [y for (x, y) in nodes] ]
    i = 1 # len(nodes)-i gives us the number of elements in the last array of p
    while len(nodes)-i > 0:
        iteration = []
        for j in range(0,len(nodes)-i):
            iteration.append( (p[-1][j+1] - p[-1][j]) / (nodes[j+i][0]-nodes[j][0]) )
        p.append(iteration)
        i = i + 1

    return p

h = 0.1
s = 0.5
nodes = [(i/10, exp(i/10)) for i in range(0,5)]
p = divided_diff(nodes)
```

This produced the following output:

$$
\begin{aligned}
f[0.0] &= 1.0 \\
f[0.0, 0.1] &= 1.051\,709\,180\,756\,477\,1 \\
f[0.0, 0.1, 0.2] &= 0.553\,046\,100\,443\,721\,5 \\
f[0.0, 0.1, 0.2, 0.3] &= 0.193\,881\,220\,406\,129\,84 \\
f[0.0, 0.1, 0.2, 0.3, 0.4] &= 0.050\,976\,664\,869\,075\,22
\end{aligned}
$$

Using these values, we get the following approximations:

$$
\begin{aligned}
P_0(0.05) &= 1.0 \\
P_1(0.05) &= 1.052\,585\,459\,037\,823\,9 \\
P_2(0.05) &= 1.051\,202\,843\,786\,714\,5 \\
P_3(0.05) &= 1.051\,275\,549\,244\,366\,8 \\
P_4(0.05) &= 1.051\,270\,770\,182\,035\,4
\end{aligned}
$$

So, our final approximation is $f(0.05) \approx P_4(0.05) = 1.0512707701820354$.

The polynomial is a Lagrange polynomial that agrees with f at the given nodes. So, the error bound is given by

$$|P_4(0.05) - f(0.05)| = \left|\frac{f^{(5)}(\xi)}{5!}\right| |(0.05 - 0)(0.05 - 0.1) \cdots (0.05 - 0.4)| < 5 \cdot 10^{-7},$$

since e^x is a strictly increasing function.