PRACTICAL-10

Newton Interpolation

PRACHI MITTAL {20211061}

B.Sc. (H) Mathematics

⊙ Computing Divided Difference

```
NDD[x0_, f0_, startindex_, endindex_] :=
  Module[{x = x0, f = f0, i = startindex, j = endindex, answer},
   If[i == j, Return[f[[i]]], answer =
      (NDD[x, f, i + 1, j] - NDD[x, f, i, j - 1]) / (x[[j]] - x[[i]]);
     Return[answer]];];
```

Ques-1

```
x = {0, 1, 3};
f = {1, 3, 55};
NDD[x, f, 1, 2]
```

```
2
```

```
x = {0, 1, 3};
f = {1, 3, 55};
NDD[x, f, 2, 3]
```

```
26
```

```
NDD[x, f, 1, 3]
```

```
8
```

Ques-2

```
x = {-1, 0, 1, 2};
f = {5, 1, 1, 11};
NDD[x, f, 1, 2]
```

```
-4
```

```
NDD[x, f, 2, 3]
```

```
0
```

```
NDD[x, f, 1, 3]
```

2

```
NDD[x, f, 2, 4]
```

5

```
NDD[x, f, 1, 4]
```

1

```
NDD[x, f, 3, 4]
```

10

⊙ Computing Polynomial

Ques-1

```
NDDP[x0_, f0_] :=
  Module[{x1 = x0, f = f0, n, newtonPolynomial, k, j},
   n = Length[x1];
   newtonPolynomial[y_] = 0;
   For[i = 1, i ≤ n, i++, prod[y_] = 1;
    For[k = 1, k ≤ i - 1, k++, prod[y_] = prod[y] * (y - x1[[k]])];
    newtonPolynomial[y_] =
     newtonPolynomial[y] + NDD[x1, f, 1, i] * prod[y]];
   Return[newtonPolynomial[y]];];
nodes = {0, 1, 3};
values = {1, 3, 55};
NDDP[nodes, values]
```

$1 + 2\,y + 8\,(-1 + y)\,y$

```
Simplify[%]
```

$1 - 6\,y + 8\,y^2$

Ques-2

```
nodes = {-1, 0, 1, 2};
values = {5, 1, 1, 11};
NDDP[nodes, values]
```

$5 - 4\,(1 + y) + 2\,y\,(1 + y) + (-1 + y)\,y\,(1 + y)$

```
Simplify[%]
```

$1 - 3\,y + 2\,y^2 + y^3$