```python
def fact(n): return 1 if n <=1 else n*fact(n-1)

class PascalTriangleRowIterator:
  def __init__(self, position, start=0):
    self.k = start
    self.n = position
    self.fact_n = fact(position)
  def __iter__(self):
    return self
  def __next__(self):
    if self.k > self.n:
      raise StopIteration
    tmp = self.fact_n//(fact(self.k)*fact(self.n-self.k))
    self.k = self.k +1
    return tmp
  def prev(self):
    if self.k <= 0:
      raise StopIteration
    self.k = self.k - 1
    return self.fact_n//(fact(self.k)*fact(self.n-self.k))

class PascalTriangleIterator:
  def __init__(self, n, start=0):
    self.dimension = n
    self.position = start
  def __next__(self):
    if self.position >= self.dimension: raise StopIteration
    self.position = self.position+1
    return PascalTriangleRowIterator(self.position-1)
  def prev(self):
    if self.position <= 0: raise StopIteration
    self.position = self.position - 1
    return PascalTriangleRowIterator(self.position,self.position+1)

class PascalTriangle:
  def __init__(self, n, start=0):
    self.rows = n
    self.start=start
  def __iter__(self):
    return PascalTriangleIterator(self.rows, self.start)
```