bottomup_rodcut.py

```
'''
Purpose : Implement a Bottom Up approach to the Rod Cutting Algorithm
Author  : Hameed Abdul
Date    : 4/1/2018
'''

import numpy as np

# Numpy array to store pipe value by length (0 - 10)
prices = np.array([0, 1, 5, 8, 9, 10, 17, 17, 20, 24, 20])

# Use to find optimal rod cutting solution, using a Bottom Up Approach
def bottomup_rodcut(prices, n):
    """
    Given the correlated prices and length for a rod, this returns the optimal rod cutting length
    to maximize profit. This is done effeciently by storing the previous results to avoid copious amounts of recalculations
    :param prices: Prices for each length
    :param n: Length to stop at
    :return: The optimal rod cutting length for each given rod
    """

    # A Numpy array from zero to N + 1
    rod_length = np.arange(n + 1)
    rod_length[0] = 0
    print('Length: \t\t\t', rod_length)

    for i in range(1, n + 1):
        q = -99999
        for j in range(1, i + 1):
            # Find the max between given price or prev sublength
            val = prices[j] + rod_length[i - j]
            q = max(q, val)
        # Set the max value and move on
        rod_length[i] = q
    return rod_length

test = bottomup_rodcut(prices, 10)
print('Prices:\t\t\t\t', prices)
print("============================================================\nMax Profit: \t\t", test)
```