1. Python Program for Topological Sorting
2. Python Program for Radix Sort
3. Python Program for Binary Insertion Sort
4. Python Program for Bitonic Sort
5. Python Program for Comb Sort
6. Python Program for Pigeonhole Sort
7. Python Program for Cocktail Sort
8. Python Program for Gnome Sort
9. Python Program for Odd-Even Sort / Brick Sort
10. Python Program for BogoSort or Permutation Sort

Here are Python programs for the sorting algorithms you requested:

### 1. Python Program for Topological Sorting

```python

from collections import defaultdict

class Graph:

def \_\_init\_\_(self, vertices):

self.graph = defaultdict(list)

self.V = vertices

def addEdge(self, u, v):

self.graph[u].append(v)

def topologicalSortUtil(self, v, visited, stack):

visited[v] = True

for i in self.graph[v]:

if not visited[i]:

self.topologicalSortUtil(i, visited, stack)

stack.insert(0, v)

def topologicalSort(self):

visited = [False] \* self.V

stack = []

for i in range(self.V):

if not visited[i]:

self.topologicalSortUtil(i, visited, stack)

print(stack)

# Example Usage

g = Graph(6)

g.addEdge(5, 2)

g.addEdge(5, 0)

g.addEdge(4, 0)

g.addEdge(4, 1)

g.addEdge(2, 3)

g.addEdge(3, 1)

g.topologicalSort()

```

### 2. Python Program for Radix Sort

```python

def countingSort(arr, exp):

n = len(arr)

output = [0] \* n

count = [0] \* 10

for i in range(n):

index = arr[i] // exp

count[index % 10] += 1

for i in range(1, 10):

count[i] += count[i - 1]

i = n - 1

while i >= 0:

index = arr[i] // exp

output[count[index % 10] - 1] = arr[i]

count[index % 10] -= 1

i -= 1

for i in range(n):

arr[i] = output[i]

def radixSort(arr):

max1 = max(arr)

exp = 1

while max1 // exp > 0:

countingSort(arr, exp)

exp \*= 10

# Example Usage

arr = [170, 45, 75, 90, 802, 24, 2, 66]

radixSort(arr)

print(arr)

```

### 3. Python Program for Binary Insertion Sort

```python

def binarySearch(arr, val, start, end):

if start == end:

if arr[start] > val:

return start

else:

return start + 1

if start > end:

return start

mid = (start + end) // 2

if arr[mid] < val:

return binarySearch(arr, val, mid + 1, end)

elif arr[mid] > val:

return binarySearch(arr, val, start, mid - 1)

else:

return mid

def binaryInsertionSort(arr):

for i in range(1, len(arr)):

val = arr[i]

j = binarySearch(arr, val, 0, i - 1)

arr = arr[:j] + [val] + arr[j:i] + arr[i + 1:]

return arr

# Example Usage

arr = [37, 23, 0, 17, 12, 72, 31]

print(binaryInsertionSort(arr))

```

### 4. Python Program for Bitonic Sort

```python

def compAndSwap(arr, i, j, dir):

if (dir == 1 and arr[i] > arr[j]) or (dir == 0 and arr[i] < arr[j]):

arr[i], arr[j] = arr[j], arr[i]

def bitonicMerge(arr, low, cnt, dir):

if cnt > 1:

k = cnt // 2

for i in range(low, low + k):

compAndSwap(arr, i, i + k, dir)

bitonicMerge(arr, low, k, dir)

bitonicMerge(arr, low + k, k, dir)

def bitonicSort(arr, low, cnt, dir):

if cnt > 1:

k = cnt // 2

bitonicSort(arr, low, k, 1)

bitonicSort(arr, low + k, k, 0)

bitonicMerge(arr, low, cnt, dir)

def sort(arr, N, up):

bitonicSort(arr, 0, N, up)

# Example Usage

arr = [3, 7, 4, 8, 6, 2, 1, 5]

N = len(arr)

up = 1

sort(arr, N, up)

print(arr)

```

### 5. Python Program for Comb Sort

```python

def getNextGap(gap):

gap = (gap \* 10) // 13

return max(gap, 1)

def combSort(arr):

n = len(arr)

gap = n

swapped = True

while gap != 1 or swapped:

gap = getNextGap(gap)

swapped = False

for i in range(0, n - gap):

if arr[i] > arr[i + gap]:

arr[i], arr[i + gap] = arr[i + gap], arr[i]

swapped = True

# Example Usage

arr = [8, 4, 1, 56, 3, -44, 23, -6, 28, 0]

combSort(arr)

print(arr)

```

### 6. Python Program for Pigeonhole Sort

```python

def pigeonholeSort(arr):

min\_val = min(arr)

max\_val = max(arr)

size = max\_val - min\_val + 1

holes = [0] \* size

for x in arr:

holes[x - min\_val] += 1

i = 0

for count in range(size):

while holes[count] > 0:

holes[count] -= 1

arr[i] = count + min\_val

i += 1

# Example Usage

arr = [8, 3, 2, 7, 4, 6, 8]

pigeonholeSort(arr)

print(arr)

```

### 7. Python Program for Cocktail Sort

```python

def cocktailSort(arr):

n = len(arr)

swapped = True

start = 0

end = n - 1

while swapped:

swapped = False

for i in range(start, end):

if arr[i] > arr[i + 1]:

arr[i], arr[i + 1] = arr[i + 1], arr[i]

swapped = True

if not swapped:

break

swapped = False

end -= 1

for i in range(end - 1, start - 1, -1):

if arr[i] > arr[i + 1]:

arr[i], arr[i + 1] = arr[i + 1], arr[i]

swapped = True

start += 1

# Example Usage

arr = [5, 1, 4, 2, 8, 0, 2]

cocktailSort(arr)

print(arr)

```

### 8. Python Program for Gnome Sort

```python

def gnomeSort(arr):

index = 0

n = len(arr)

while index < n:

if index == 0:

index += 1

if arr[index] >= arr[index - 1]:

index += 1

else:

arr[index], arr[index - 1] = arr[index - 1], arr[index]

index -= 1

# Example Usage

arr = [34, 2, 10, -9]

gnomeSort(arr)

print(arr)

```

### 9. Python Program for Odd-Even Sort (Brick Sort)

```python

def oddEvenSort(arr):

n = len(arr)

isSorted = False

while not isSorted:

isSorted = True

for i in range(1, n - 1, 2):

if arr[i] > arr[i + 1]:

arr[i], arr[i + 1] = arr[i + 1], arr[i]

isSorted = False

for i in range(0, n - 1, 2):

if arr[i] > arr[i + 1]:

arr[i], arr[i + 1] = arr[i + 1], arr[i]

isSorted = False

# Example Usage

arr = [34, 2, 10, -9]

oddEvenSort(arr)

print(arr)

```

### 10. Python Program for BogoSort or Permutation Sort

```python

import random

def is\_sorted(arr):

for i in range(len(arr) - 1):

if arr[i] > arr[i + 1]:

return False

return True

def bogoSort(arr):

while not is\_sorted(arr):

random.shuffle(arr)

# Example Usage

arr = [3, 2, 5, 1, 0, 4]

bogoSort(arr)

print(arr)

```

These programs cover a wide range of sorting algorithms from the standard ones to more obscure types. Let me know if you need any more details or further explanations!