**Cocktail Shaker Sort**

**Cocktail Shaker Sort:**

* An improved variation of bubble sort
* It **traverses a list of elements both forwards and backwards** instead of only forwards, and **swaps values if needed**

**Cocktail Shaker Sort Steps:**

**def cocktail\_shaker\_sort(a):**

**swap = True**

**ti = 0**

**while swap:**

**swap = False**

**for i in range(len(a)-1):**

**if a[ti] > a[ti+1]:**

**swap = True**

**a[ti], a[ti+1] = a[ti+1], a[ti]**

**ti += 1**

**if not swap:**

**break**

**for j in range(len(a), -1)):**

**if a[ti] < a[ti-1]:**

**swap = True**

**a[ti], a[ti-1] = a[ti-1], a[ti]**

**ti -= 1**

* **Very similar to Odd-Even-Sort**

**Time and Space Complexity**

**Time Complexity Space Complexity**

**Best Average Worst Best Average Worst**

**O(n) O(n\*\*2) O(n\*\*2) O(1)**

**This algorithm is still pretty inefficient, and does NOT get used in many practical, real world, applications.**