Yazılım Geliştirme Laboratuvarı

Sıralama Algoritması Görselleştiricisi

201307031   
Yiğit Taşkın  
taskinyigit84@gmail.com

201307017  
Ebru Merdtuanaebru81@gmail.com

201307073  
Dilara Taşkırandilarataskiran0@gmail.com

*Abstract* : *Embedding sorts refers to those used to organize an array element according to a certain criteria or rule. Carry out the sorting work of these descriptions by comparing the items. We visualized these images using Python and tkinter.*

*Keywords: Python , Algorithms,Tkinter*

*Özet : yerleştirmeli sıralamaları, bir dizi öğesini belirli bir kriter veya kurala göre düzenleme için kullanılanları ifade eder. Bu açıklamaları, öğeleri karşılaştırarak sıralama çalışmasını yürütün. Python ve tkinter kullanarak bu görüntüleri görselleştirdik.*

*Anahtar Kelimeler: Python, Algoritma, Tkinter*

# GİRİŞ

Sıralama algoritmaları, bir dizi öğeyi belirli bir kriter veya kurala göre düzenlemek için kullanılan algoritmaları ifade eder. Bu algoritmalar, çeşitli uygulamalarda kullanılır, örneğin veri analizi, veritabanı sorgulama, grafik algoritmaları, arama motorları ve daha fazlası.

Karşılaştırmalı Sıralama Algoritmaları:

Karşılaştırmalı sıralama algoritmaları, sıralanacak öğeleri karşılaştırarak sıralama işlemini gerçekleştirir. Bu algoritmalar, genellikle öğelerin küçükten büyüğe veya büyükten küçüğe doğru sıralanmasında kullanılır.

* Seçme Sıralaması (Selection Sort): En küçük öğeyi bulup ilk sıraya yerleştirir, ardından ikinci en küçük öğeyi bulup ikinci sıraya yerleştirir ve bu işlemi devam ettirerek sıralama yapar.
* Ekleme Sıralaması (Insertion Sort): Bir sonraki öğeyi sıralı kısmın uygun konumuna yerleştirir. Her adımda, bir önceki sıralı kısım genişler ve yeni öğe doğru konumuna yerleştirilir.
* Hızlı Sıralama (Quick Sort): Bir pivot eleman seçer ve pivot elemandan küçük olanları soluna, büyük olanları ise sağa yerleştirir. Ardından, sol ve sağ bölümlere ayrı ayrı aynı işlemi uygulayarak sıralama yapar. Veri kümesi parçalara ayrıldıkça sıralanır.
* Birleştirme Sıralaması (Merge Sort): Veri kümesini rekürsif olarak parçalara ayırır, ardından parçaları birleştirerek sıralar. Parçalar, birleştirme işlemi sırasında karşılaştırılıp birleştirilir.

# SIRALAMA ALGORİTMALARI

**Kabarcık Sıralaması (Bubble Sort):** Komşu öğeleri karşılaştırarak sıralama yapar ve en büyük öğeleri sağa doğru iter. Yavaş bir sıralama algoritmasıdır ve genellikle küçük veri setleri için tercih edilir.

def bubble\_sort(arr):

n = len(arr)

for i in range(n-1):

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

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

arr[j], arr[j+1] = arr[j+1], arr[j]

**Seçme Sıralaması (Selection Sort):** En küçük öğeyi bulup ilk sıraya yerleştirir, ardından ikinci en küçük öğeyi bulup ikinci sıraya yerleştirir ve bu işlemi devam ettirerek sıralama yapar.

def selection\_sort(arr):

n = len(arr)

for i in range(n):

min\_idx = i

for j in range(i+1, n):

if arr[j] < arr[min\_idx]:

min\_idx = j

arr[i], arr[min\_idx] = arr[min\_idx], arr[i]

**Ekleme Sıralaması (Insertion Sort):** Bir sonraki öğeyi sıralı kısmın uygun konumuna yerleştirir. Her adımda, bir önceki sıralı kısım genişler ve yeni öğe doğru konumuna yerleştirilir.

def insertion\_sort(arr):

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

key = arr[i]

j = i - 1

while j >= 0 and arr[j] > key:

arr[j + 1] = arr[j]

j -= 1

arr[j + 1] = key

**Hızlı Sıralama (Quick Sort):** Bir pivot eleman seçer ve pivot elemandan küçük olanları soluna, büyük olanları ise sağa yerleştirir. Ardından, sol ve sağ bölümlere ayrı ayrı aynı işlemi uygulayarak sıralama yapar. Veri kümesi parçalara ayrıldıkça sıralanır.

def quick\_sort(arr):

if len(arr) <= 1:

return arr

pivot = arr[len(arr)//2]

left = [x for x in arr if x < pivot]

middle = [x for x in arr if x == pivot]

right = [x for x in arr if x > pivot]

return quick\_sort(left) + middle + quick\_sort(right)

**Birleştirme Sıralaması (Merge Sort):** Veri kümesini rekürsif olarak parçalara ayırır, ardından parçaları birleştirerek sıralar. Parçalar, birleştirme işlemi sırasında karşılaştırılıp birleştirilir..

def merge\_sort(arr):

if len(arr) <= 1:

return arr

mid = len(arr) // 2

left = arr[:mid]

right = arr[mid:]

left = merge\_sort(left)

right = merge\_sort(right)

return merge(left, right)

def merge(left, right):

result = []

i = j = 0

while i < len(left) and j < len(right):

if left[i] < right[j]:

result.append(left[i])

i += 1

else:

result.append(right[j])

j += 1

result.extend(left[i:])

result.extend(right[j:])

return result

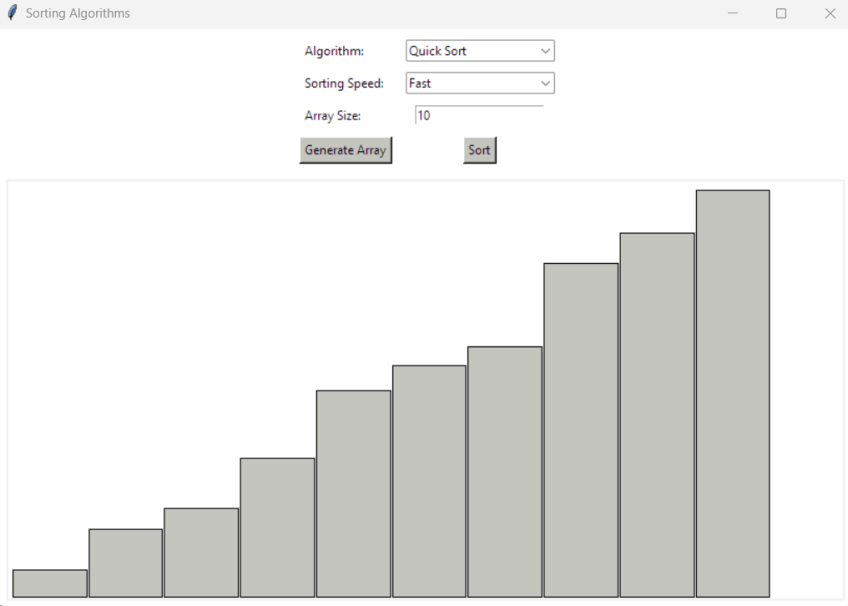
# KULLANILAN TEKNOLOJİLER

Python, yüksek seviyeli bir programlama dilidir ve genel amaçlı bir dil olarak kullanılır. İngiliz programcı Guido van Rossum tarafından geliştirilen Python, basit ve anlaşılır sözdizimiyle bilinir. Python, çeşitli programlama alanlarında kullanılan geniş bir kütüphane ve araç setine sahiptir.

Tkinter, Python programlama dilinin standart kütüphanelerinden biridir. Tkinter, Python ile masaüstü uygulamaları geliştirmek için kullanılan bir GUI (Graphical User Interface - Grafiksel Kullanıcı Arayüzü) aracıdır. Tkinter, Tk GUI toolkit'inin Python diline bağlanmış halidir. Tk, popüler bir GUI toolkit'idir ve Tkinter üzerinden Python ile etkileşimli ve görsel uygulamalar geliştirmek için kullanılır.

Tkinter, kullanıcıya düğmeler, metin kutuları, etiketler, listeler ve diğer kullanıcı arayüzü öğeleri gibi grafiksel bileşenler sunar. Bu bileşenler, kullanıcıların uygulamalarla etkileşimde bulunmasını sağlar. Tkinter, platformlar arası uyumluluğa sahiptir ve Windows, macOS ve Linux gibi farklı işletim sistemlerinde çalışabilir.

# PROJE GÖRSELLERİ



##### 

##### ReferANSLAR

1. https://youtu.be/m3IxvuN0NV8J. Algoritma Dersleri
2. https://youtu.be/ioNfInaP-EA
3. <https://docs.python.org/3/library/tkinter.html>
4. <https://realpython.com/python-gui-tkinter/>

***GİTHUB LİNKİ :*** <https://github.com/yigittaskin/py_sorting_algorithms>

1. R. Nicole, “Title of paper with only first word capitalized,” J. Name Stand. Abbrev., in press.
2. Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, “Electron spectroscopy studies on magneto-optical media and plastic substrate interface,” IEEE Transl. J. Magn. Japan, vol. 2, pp. 740–741, August 1987 [Digests 9th Annual Conf. Magnetics Japan, p. 301, 1982].
3. M. Young, The Technical Writer’s Handbook. Mill Valley, CA: University Science, 1989.

**IEEE conference templates contain guidance text for**

**composing and formatting conference papers. Please ensure that all template text is removed from your conference paper prior to submission to the conference. Failure t**