Algoritma Analizi ve Tasarımı Dersi

2022-2023 Bahar Dönemi

**ÖDEV RAPORU**

Teslim Tarihi: 15.05.2023

Öğrenci Adı: İlayda DİNÇBİLEK

Öğrenci Numarası: 212802040

1. Giriş

Ödevin amacı, 0/1 Knapsack problemi için bir algoritma geliştirmektir. Algoritma, giriş olarak bir dosya adı alacak ve bu dosyada tanımlanan öğelerin değerlerini, ağırlıklarını ve çanta kapasitesini kullanarak en değerli kombinasyonu bulacaktır. Çıkış olarak, maksimum toplam değeri, seçilen öğelerin listesini ve bu öğelerin değerlerini içeren bir çıktı sağlanacaktır. Ayrıca, bu sonuçlar bir Excel dosyasına kaydedilecektir.

1. Geliştirme ve Çalıştırma Ortamları

Algoritma Visual Studio Code ortamında Python dilinde yazılmıştır.

3. Algoritma Akışı

1. Dosya adını al

2. Dosyayı aç ve oku

3. İlk satırı oku ve num\_items ve total\_weight değişkenlerine ata

4. Boş values ve weights listeleri oluştur

5. lines[1:] satırlarını döngüyle işle

1. Satırdaki v ve w değerlerini oku ve values ve weights listelerine ekle

6. total\_weight+1 ve num\_items+1 boyutunda 0'lardan oluşan bir tablo oluştur

7. 1'den num\_items+1'e kadar bir döngü oluştur

1. 1'den total\_weight+1'e kadar bir döngü oluştur

1. Eğer i-1. öğenin ağırlığı j'den büyükse, table[i][j] = table[i-1][j]

2. Değilse, table[i][j] = max(table[i-1][j], table[i-1][j-weights[i-1]] + values[i-1])

8. table[num\_items][total\_weight] değerini max\_value olarak ata

9. Seçilmiş öğeleri saklamak için num\_items boyutunda 0'lardan oluşan selected\_items ve selected\_item\_values listeleri oluştur

10. j = total\_weight ve num\_items'tan 1'e kadar bir döngü oluştur, adımı -1 olarak ayarla

1. Eğer table[i][j] != table[i-1][j] ise,

1. selected\_item\_values[i-1] = values[i-1] olarak ayarla

2. selected\_items[i-1] = 1 olarak ayarla

3. j -= weights[i-1]

11. selected\_item\_values'da sıfır olmayan tüm değerleri seçilen öğe değerleri olarak sakla

12. max\_value, selected\_items ve selected\_item\_values değerlerini döndür

13. Bir dizi dosya adı ve ilgili öğe sayısı oluştur

14. Bir Excel dosyası oluştur ve verileri yazdır

15. Çalışma süresini hesapla ve ekrana yazdır

16. x ve y dizilerini oluştur

17. Grafikleri oluştur

18. Excel dosyasını kaydet

1. Sonuç

* Knapsack problemi, sınırlı kapasiteli bir çantaya (veya knapsack) sığdırılabilecek en değerli öğeleri seçme problemidir. Bu problem, çeşitli alanlarda kullanılan bir optimizasyon problemidir. Genellikle üretim, dağıtım, yatırım, depolama, reklamcılık, seyahat planlama gibi alanlarda kullanılır.
* Örneğin, üretimde hammaddelerin sınırlı stokları varsa, knapsack algoritması kullanılarak hangi hammaddelerin kullanılacağı ve hangi miktarlarda kullanılacağı belirlenebilir. Benzer şekilde, seyahat planlama uygulamalarında seyahat edilecek yerlerin sınırlı zaman ve bütçe ile seçilmesi için knapsack algoritması kullanılabilir. Ayrıca, finansal portföy yönetiminde yatırım yapılacak finansal varlıkların sınırlı sayıda olması durumunda, knapsack problemi kullanılarak en verimli portföy seçimi yapılabilir.