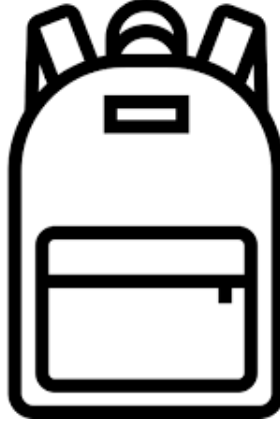


CENG 112 - Final Ödevi

KNAPSACK PROBLEMİ

Teslim Tarihi: 10.06.2020

Bu final ödevinde bilgisayar bilimlerinin meşhur *Knapsack problemi*ni çözmeye çalışacağız. Bu problemde



belirli bir kapasiteye sahip bir sırt çantası ve bu çantanın içerisine girebilecek elemanlar (item) bulunmaktadır. Bu elemanların da kendine ait ağırlığı ve de değeri vardır. Bu problemdeki amaç çantanın kapasitesini geçmeyecek şekilde toplamda en yüksek değere sahip olan elemanları yerleştirebilmektir. İstenen sınıflar Knapsack.java ve Item.java içerisinde gerçekleştirilmelidir.

a. Knapsack sınıfı için

- Çantanın kapasitesi(ağırlık olarak): - double tipinde
- Çantanın içerisindeki elemanlar: + ArrayList<Item> tipinde
- public Knapsack(double capacity): constructor'ı, gerekli initializationlar da yapılmalı.
- public double getCapacity();
- public void setCapacity(double);
- public ArrayList<Item> getItems();
- public void setItems(ArrayList<Item>); Çantanın içerisinde o anda bulunan elemanları ArrayList içerisinde döndürmektedir.
- public boolean pushItem(Item item): Eğer çantanın içersinde parametre olarak gönderilen elemanın ağırlığını alacak kapasite kaldıysa elemanlar ArrayListinin içerisine eklenmeli ve true gönderilmelidir. Eğer kapasite yetmiyorsa false döndürmelidir.
- public double getMaximumValue(ArrayList<Item> newItems): Çantada boş yer varsa kalan yere maksimum değeri oluşturacak elemanların yerleştirilmesi ve sonuçta çantada bulunan tüm elemanların toplam değerini döndüren metottur (varsa ilk baştaki bulunanlar da dahil).
- public ArrayList<Item> getMaximalItemSet(ArrayList<Item> newItems): Çantada boş yer varsa kalan yere maksimum değeri oluşturacak elemanların yerleştirilmesi ve sonuçta çantada bulunan tüm elemanları döndüren metottur (varsa ilk baştaki bulunanlar da dahil).

b. Item sınıfı: (Comparable interface'ini implement etmeli)

- Bu sınıf comparable Interface'ini implement etmelidir.

- **public int compareTo (Object itemOb):**
- elemanın değerini tutan alan: -double
- elemanın ağırlığını tutan alan:- double
- public double getWeight():
- public void setWeight(double weight)
- public double getValue(): -
- public void setValue(double weight): constructor'ı
- public Item (double value, double weight): Constructor
- public int compareTo(Object o)
- public ArrayList<Item> sort(ArrayList<Item> itemL): parametre olarak gelen elemanları önce değerine göre, değeri aynı ise ağırlığına göre küçükten büyüğe göre sıralamalı ve sıralanmış listeyi döndürmelidir.

Gerekirse başka metotlar da ekleyebilirsiniz.

Ödevleriniz hem bilmoodle sistemi üzerinden hem de EDS üzerinden toplanacaktır. Bilmoodle üzerinden, yazdığınız kodların kontrolü, sistem üzerinden evaluate butonuna basmanız ile gerçekleştirilecektir. EDS için ise; yazdığımız kodların yanı sıra ayrıca bir rapor oluşturmanız gerekmektedir. Kodlarınızın bulunduğu klasörün ismi öğrenci numaranız olmalıdır. Raporun başında; İsim, soyisim ve Öğrenci numarası unutulmamalıdır. Rapor için oluşturduğunuz pdf'in isimlendirilmesi **İsim-Soyisim-Öğrno.pdf** olarak yapılmalıdır. **Aynı ödevi gönderdiği tespit edilen kişiler hakkında tutanak tutulacaktır.** Raporda yazdığınız kodları sınıf ve metot bazında anlatan açıklamalar bulunmalıdır. Sistemde tam puan almayan bir metodunuz varsa kodunu rapora okunaklı bir şekilde ekleyip belirtiniz.

Hem rapor hem de kod dosyalarınız **İsim-Soyisim-Öğrno.rar** olarak tek bir dosya şeklinde EDS üzerine yüklenmelidir. Her iki sisteme yüklenmeyen dosyalar değerlendirmeye alınmayacaktır. Ayrıca ödevin son gönderim tarihinden sonra herhangi bir ödev kabul edilmeyecektir.

Teşekkür ederiz.

Dr. Fatmana ŞENTÜRK & Dr. Elif HAYTAOĞLU