YAPAY ZEKA VE PYTHON PROGRAMLAMA

Kümeler ve Sözlükler
Emir Öztürk

Dr. Öğr. Üyesi Altan MESUT'un "Programlama Dillerine Giriş" ders slaytları kaynak alınmıştır.

KÜME (SET)

- Tekrarsız elemanların listesi
- Birleşim
- Kesişim
- Fark
- İndis desteği yok.
- Tanımı küme parantezleri ile yapılır.

Listeler gibi ilgili verileri aynı yapı içinde saklamayı amaçlayan kümelerin temel farkı; aynı elemanın bir küme içinde yalnızca bir defa yer alabilmesidir.
Aslında matematikteki küme kavramı ile benzerdir ve kümeler arasında birleşim, kesişim ve fark gibi işlemler yapılabilir.
Listeler, demetler ve string'lerde olduğu gibi indis verilerek elemanlarına ulaşılamaz. Çünkü elemanlar belirli bir sıra ile kaydedilmez.
Tanımlanırken elemanlar küme parantezleri içinde verilir.

ÖRNEK

- a = {1,2,3,4,5}
- b = {3,4,5,6,7,8}
- a | b = ?
- a & b = ?
- a - b = ?
- a ^ b = ?

KÜME FONKSİYONLARI

- küme1.union(küme2) (|)
- küme1.update(küme2) (|=)
- küme1.intersection(küme2) (&)
- küme1.intersection_update(küme2) (&=)
- küme1.difference(küme2) (-)
- küme1.difference_update(küme2) (-=)
- küme1.symmetric_difference(küme2) (^)
- küme1.symmetric_difference_update(küme2) (^=)

union iki kümenin birleşimi için kullanılır.
intersection kümelerin kesişimini verir.
difference iki kümenin farkını verir.
symmetric_difference iki kümenin kesişimi dışında kalan değerleri verir.
tüm update metotları başına belirtilen işlem ile elde edilen sonucun ilk kümeye aktarılması işlemini gerçekleştirir.

KÜMELERLE İLGİLİ KARŞILAŞTIRMA YÖNTEMLERİ

- küme1.isdisjoint(küme2) (!=)
- küme1.issubset(küme2) (<=)
- küme1.issuperset(küme2) (>=)
- in operatörü (Eleman içerme kontrolü)

küme1.isdisjoint(t) s != t Eğer s kümesi ile t kümesinin kesişimi boş ise True döndürür
küme1.issubset(t) s <= t Eğer s kümesi t kümesinin alt kümesi ise True döndürür
küme1.issuperset(t) s >= t Eğer s kümesi t kümesini kapsıyorsa True döndürür
IN OPERATÖRÜ: for döngüsünde range öncesi veya liste, demet, string gibi bir iterable yapı öncesi kullanılan ‘in’ ifadesi aslında bir karşılaştırma operatörüdür (yani boolean değer üretir). Kümeler ile de kullanılabilir: x in s işleminde eğer x elemanı s kümesinin elemanı ise True, değilse False döner. Eğer ‘not in’ yazılırsa tersi olur. Örnek: 35 in range(1,100,2) → True

KÜME FONKSİYONLARI

- küme.add(eleman)
- küme.remove(eleman)
- küme.discard(eleman)
- küme.pop()
- küme.clear()
- küme.copy()

küme1.add(x): x elemanını s kümesine ekler. Eğer bu eleman zaten varsa eklemez (hata da vermez).
küme1.remove(x): s kümesinden x elemanını çıkarır. Eğer eleman kümede yoksa hata verir.
küme1.discard(x): s kümesinde x elemanı var ise çıkarır. Yoksa hata vermez.
küme1.pop(): s kümesinden keyfi bir eleman siler ve bu elemanı döndürür.
küme1.clear(): Kümenin tüm elemanlarını siler.
küme1.copy(): Kümenin kopyasını döndürür.
Eğer s bir küme ise; t = s kullanımı t ile s'nin aynı kümeyi göstermesini sağlar. t = s.copy() kullanımı ise t'yi s'nin kopyası olan yeni bir küme yapar. Yani birinin elemanlarının değişmesi diğerini etkilemez (Aynı durum listedeki copy için de gerçerli).

ÖRNEKLER

- küme = set("tekrarlı")
- küme içeriği?
- sonuc = {x for x in a if x not in b}
- sonuc ?
- küme >= sonuc ?

SÖZLÜK

- Anahtar - değer
- Anahtar tekil
- İndis olarak anahtar
- Varsa değiştir, yoksa ekle

Sözlük, anahtar-değer (key-value) ikililerinin bir kümesidir. Değer tekrar edebilir ama Anahtar tekil (unique) olmalıdır, yani sözlük içinde iki defa aynı anahtar bulunamaz (kümelerde aynı elemanın iki defa bulunamaması gibi).
Kümelerden farklı olarak indis verilerek elemanlarına ulaşılabilir. İndis olarak ta anahtar kullanılır. Sözlükteki belli bir indise (yani anahtara) değer atanmak istendiğinde, eğer o indise ait bir değer varsa yeni değer ile değiştirilir. Yoksa, sözlüğe yeni bir anahtar-değer ikilisi eklenir.

SÖZLÜK

- Küme parantezleri ile tanım
- anahtar değer aralarında ":" karakteri
- elemanlar arasında ","
- sozluk = { } ?
- Boş set için set()

Sözlükler de kümeler gibi küme parantezleri içinde tanımlanır. Anahtar ve değer arasında ':' kullanılır. Her eleman arasında yine ',' kullanılır.
D = {} kullanımı D'yi boş bir sözlük olarak tanımlar.
D'nin boş küme olması için D = set() kullanılır.

SÖZLÜK METOTLARI

- sözlük.get(anahtar,varsayılan)
- sözlük.setdefault(anahtar,değer)
- sözlük.pop(anahtar,varsayılan)

sozluk.get(k, m): Eğer k anahtar olarak sözlükte varsa, ilişkili değeri döndürür. Yoksa verilen m değerini döndürür (bu değer verilmediyse None döndürür). İndis kullanımı ile de aynı işlemi yapabilirsiniz ama anahtar bulunamazsa hata alırsınız.
sozluk.setdefault(k, m): Eğer k anahtar olarak sözlükte varsa, ilişkili değeri döndürür. Yoksa yeni bir eleman olarak ekler ve m parametresini değer olarak kullanır (verilmediyse None kullanır).
sozluk.pop(k, m): 'get' ile aynı işi yapar ama sözlükten ilgili elemanı da siler. Eğer k sözlükte anahtar olarak yoksa KeyError hatası verir (del komutu da aynı)

SÖZLÜK METOTLARI

- sözlük.popitem()
- sözlük.update(sözlük2)
- sözlük.fromkeys(anahtarlistesi)
- sözlük.items()
- sözlük.keys()
- sözlük.values()

sozluk.popitem(): Kümelerdeki pop gibi keyfi bir eleman siler ve bu elemanı döndürür.
sozluk.update(s): s sözlüğündeki anahtarlarla aynı olan d sözlüğünde anahtarlar varsa d sözlüğündeki değerleri s sözlüğündekiler ile günceller. Bulunamayan anahtarlar için anahtar-değer ikililerini de d sözlüğüne ekler. s bir sözlük yerine anahtar-değer çiftlerini içeren bir iterable olabilir. s verilmezse d aynı kalır.
sozluk.fromkeys(s, m): içine aldığı s iterable'ı ile anahtarları oluşturarak yeni bir sözlük döndürür. Seçimlik olan m ile her anahtar için sabit bir başlangıç değeri verilebilir (verilmezse None kullanır)
sozluk.items(), sozluk.keys() ve sozluk.values(): Herhangi bir parametre almayan bu yöntemler sözlükteki anahtar-değer ikililerini (demet şeklinde), sadece anahtarları veya sadece değerleri bir liste olarak elde etmeyi sağlayan görünüm nesneleri (view objects) döndürür.

SORTED

- sorted(elemanlar)
- elemanlar: demet, liste, string, küme, sözlük
- key, reverse

sorted(iterable) fonksiyonu bir iterable (demet, liste, string, küme, sözlük) alır ve elemanlarını sıralayarak bir liste olarak döndürür.
Geçen hafta verilen sort yönteminde olduğu gibi seçimlik olarak key= (varsayılan None) ve reverse= (varsayılan False) parametrelerini alır.
Listelerde zaten sort yöntemi olmasına rağmen, eğer listeyi sadece sıralı olarak görmek, mevcut şeklini değiştirmemek isteniyorsa bu fonksiyona parametre olarak o liste verilebilir.

ÖRNEK SORULAR

- Rastgele n adet sayı tutup ekrana gösteren uygulama
- Rastgele n adet birbirinden farklı sayı tutup ekrana gösteren uygulama
- Rastgele 1-10 arası n adet sayı tutup hangi sayıdan kaç adet geçtiğini gösteren uygulama
- Girilen ad soyad ve yaş bilgilerine göre oluşturulan bir liste içerisinden rastgele 5 kişi seçilecektir. Seçilen kişinin 18 yaşından küçük olması durumunda bu 5 kişi içerisine dahil edilmeyecektir ve çekiliş işlemi tekrarlanacaktır.