



Kavramlar

Algoritma:
Bilgisayarda bir *problemin çözümü için oluşturulan mantıksal akış sırasıdır*. Problemin çözülmesi için geliştirilmiş yöntemdir.

```

    Giriş (VERİ) → ALGORİTMA → Çıkış (Sonuç)
  
```

Programlama ve Program:
Oluşturulan *algoritmanın bir programlama dili ile (Java, C, C#...) yazılması* programlama, programlama dili ile ifade edilmiş hali **program** adını alır.

Dr. Sinan TUNCEL

Kavramlar

Veri:
Programlar tarafından *işlenen en temel elemandır*.
Örnek: sayısal, metinsel bilgiler, resimler, sesler, girdi/çıkış olarak veya hesaplamalarda kullanılan bilgiler....

Veri Yapıları:
Verilerin düzenlenme biçimini belirleyen yapıtaşlarıdır (değişkenler en basit veri yapısıdır) .

Programlar içinde kullanılan algoritmalarda verilerin diziler, yığınlar, kuyruklar, ağaçlar ve çizgeler gibi isimler verilmiş veri yapıları şeklinde düzenlenmesi gerekebilir.

Dr. Sinan TUNCEL

Kavramlar

Temel Veri Yapıları:
Daha çok programlama dilleri tarafından doğrudan değişken veya sabit bildirimi yapılarak kullanılır.

Her programlama dilinin temel veri türleri arasında farklılıklar olsa da tamsayı, karakter, gerçel (kesirli) sayı gibi türler hemen hemen her programlama dilinde vardır.

Karakterler
ASCII Her karakter 8 bit ($2^8 = 256$ farklı karakter)
Unicode Her karakter 16 bit ($2^{16} = 65536$ farklı karakter)

Tamsayılar
8 bit short, short int, ShortInt, byte
16 bit integer, int, integer16, Int16
32 bit long, long int, LongInt, integer32, Int32

Ondalıklı (Gerçel) Sayılar
16 bit half (IEEE 754-2008)
32 bit float (C)
64 bit double

Dr. Sinan TUNCEL

Kavramlar

Diziler: Karakter dizileri , Sayı dizileri (**Array**) (Tek boyutlu, İki boyutlu (matris), Çok boyutlu)

Tanımlanmalı veri yapıları
Kendisinden önceki tanımlanmalı veya temel veri yapıları üzerine kururlar; yani, önceden geçerli olan veri yapıları kullanılarak sonradan tanımlanırlar.

Struct
Enumeration

Veri Yapıları : Liste, Yığıt, Kuyruk, Ağaç, Çiz

Not: Veri Tabanı (Database) : işlenecek / işlenmiş verilerden oluşan bilgi bankası

Dr. Sinan TUNCEL

Veri - Yapı - Algoritma Arasındaki İlişki

Bir programda *veri, yapı ve algoritma* birbirinden ayrılmaz bileşenlerdir ve her biri önemlidir. Yapı iyi tasarlandığında, etkin, doğru, anlaşılır, hızlı ve az kaynak kullanan programlar geliştirilir.

Bir programın daha etkin, anlaşılır ve doğru olabilmesi için, algoritmanın geliştirilmesi aşamasında verilerin düzenlenmesi gerekir.

Veri yapıları, sayısal sistemlerin içyapısında bit dizisi şeklinde bulunan verilerin anlamlı hale getirilmesi için kullanılan bir biçimleme / kalıplama yöntemidir.

Örneğin 16 tane bit bir araya getirilip tamsayı (integer) diye bir veri türü oluşturulur; 8 tane bit bir araya getirilip harflerin ve özel simgelerin tutulabileceği karakter veri türü oluşturulur. Buna benzer, tamsayı, karakter gibi program tasarımı sırasında çok fazla kullanılan birçok tanımlama yapıları ve temel veri türleri ortaya koyulur:

Dr. Sinan TUNCEL

Aynı bit dizisi, verinin formatına bağlı olarak farklı anlamlara gelebilir,

0100 0010 0100 0001 0100 0010 0100 0001

4 2 4 1 4 2 4 1

Yukarıdaki bit dizisi;

Karakter dizisi (string) ise (ASCII): B A B A

BCD (Binary Coded Decimal) ise: 4 2 4 1 4 2 4 1

16-bit tam sayı ise: 16961 16961

32-bit tam sayı ise: 1111573057

32-bit gerçel sayı ise: 0.4276801x1066

Dr. Sıgır TUNCEL 7

Veri Yapılarının/Modellerinin Kıyaslanması

Veri Yapısı	Artıları	Eksileri
Dizi	Hızlı ekleme ve çok hızlı erişim (index biliniyorsa).	Yavaş arama, yavaş silme ve sabit boyut.
Sıralı Dizi	Sıralanmamış diziye göre daha hızlı arama.	Yavaş arama, yavaş silme ve sabit boyut.
Yığın	Son giren, ilk çıkar (last-in, first-out) erişimi sağlar.	Diğer öğelere yavaş erişim.
Kuyruk	İlk giren, ilk çıkar (first-in, first-out) erişimi sağlar.	Diğer öğelere yavaş erişim.
Bağlı Liste	Hızlı ekleme ve silme.	Yavaş arama.
Hash Tablosu	Hızlı ekleme ve anahtar bilindiğinde çok hızlı erişim.	Yavaş silme, anahtar bilinmediğinde yavaş erişim ve verimsiz bellek kullanımı.
Küme(Heap)	Hızlı ekleme ve silme.	Diğer öğelere yavaş erişim. Başta en büyük öğeye erişim.
İkili Ağaç	Hızlı arama, ekleme ve silme (ağaç dengeli kalmışsa).	Silme algoritması karmaşık.
Graf	Gerçek-dünya problemlerini modelleyebilmesi.	Bazı algoritmaları yavaş çalışmakta ve karmaşıklığı yüksek.

Dr. Sıgır TUNCEL 8

İyi çalışmalar...

Dr. Sıgır TUNCEL 9