Veri Yapıları ve Algoritmalar
Temel Kavramlar

Dr. Yunus Santur

Temel Tanımlar

Veri: Algoritma tarafından işlenen ham enformasyon parçalarıdır.

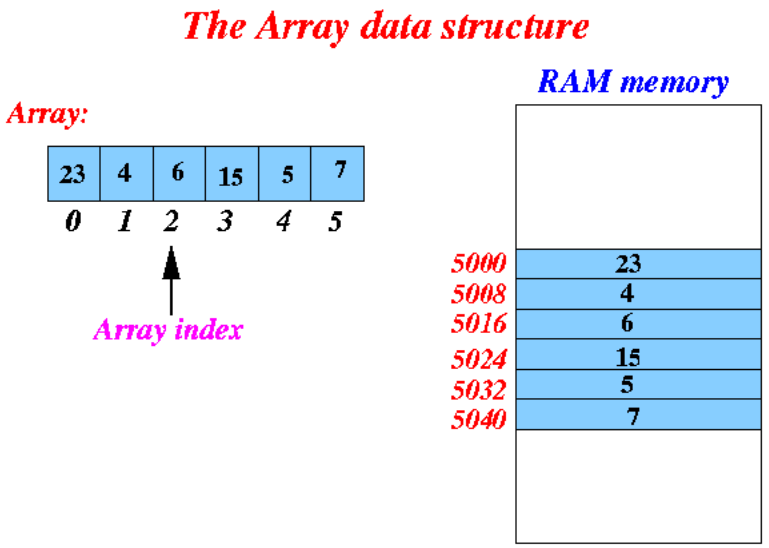
Veri Yapısı: Verilerin bellekte saklanma şeklini ifade eder.

Veri Modeli: Verilerin ilişkisel düzenini ifade eder.

Veri Tabanı: verilerin belirli kurallar dahilinde saklanması, gerektiğinde hızlı bir şekilde aranıp bulunmasını sağlayan bir saklama/sorgulama yazılımıdır.

Örnek

- Aşağıdaki şekilde dizilerin veri yapısı/modeli verilmektedir.



Temel Kavramlar

- Algoritma: Problemin çözümü için izlenmesi gereken adımlar bütünü

- Program: Algoritmanın bir programlama dili ile kodlanmış halidir.

Bir program kendi içinde bir bütün olarak çalışan ve belirli bir problemi çözen yazılımın bir parçasıdır.

Algoritma/Programa ait bilinmesi gerekenler

1. Programın çalışma hızı
   Big-O
   Benchmark

2. Programın bellek gereksinimi

Temel Veri Yapıları

- Her programlama dilinde aşağıdaki temel veri türleri mevcuttur.
  - Tamsayı (integer/long)
  - Karakter (char)
  - Rasyonel sayı (double/float)
  - Sözcük (String)
  - Diziler (Ardışıl veri türleri)

- Bazı programlama dillerinde sözlük, karmaşık sayı gibi veri türleri de temel veri tipi olarak kabul edilebilmektedir.
  - Python (sözlük)
  - Fortran (karmaşık sayı)
  - Lisp (ağaç)

Karakter veri yapısı

- ASCII kod tablosu:
  - Standart ascii: 7-bit, 128 karakter
  - Genişletilmiş ascii: 8-bit, 255 karakter

- Unicode: 16-bit

Farklı dillerde ki (Türkçe/Çince) karakterlerin gösterimi için geliştirilmiştir.

Tamsayı

İkili sayı sistemi bilinmelidir.

n-bit ile 2^n durum temsil edilebilir.

Örnek: 8-bit veri yapımız var ise 2^8=128 durum temsil edebiliriz.

Böylece 0..127 arasındaki sayılar için 7-bitlik bir veri yapısı kullanabiliriz.

Tamsayı

- Peki ya negatif sayılar?

N-bitlik verideki en anlamlı hane (sol baştaki) işaret biti olarak ayrılır.

İşaret biti 0 ise sayı pozitif, 1 ise sayı negatiftir (n-1) bit ise sayının kendisi için ayrılır.

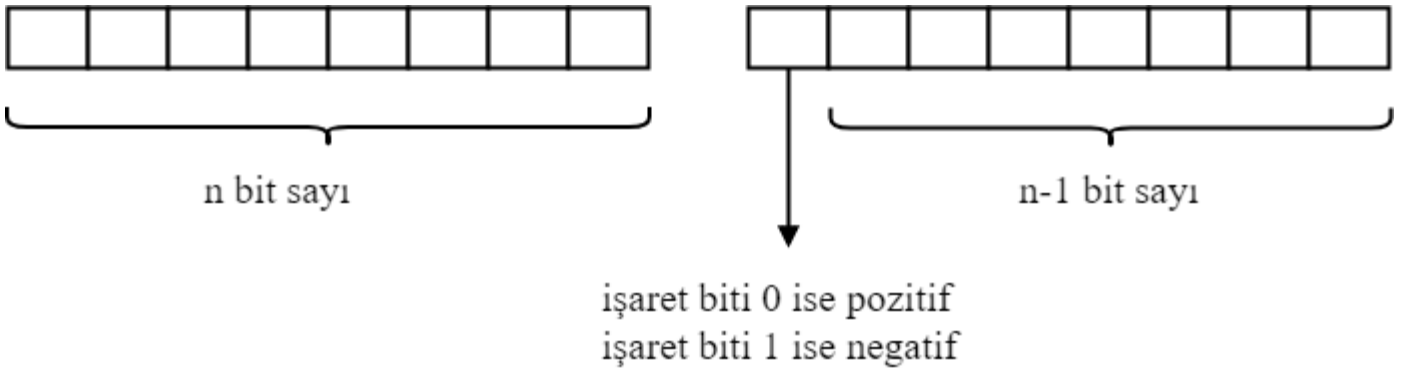
Önceki örneğe dönersek, 8-bitlik veri ile 2^7=128 durum temsil edebiliriz.

-128…-1 , 0…127 (1-bit işaret biti olarak ayrıldı!)

Tamsayı

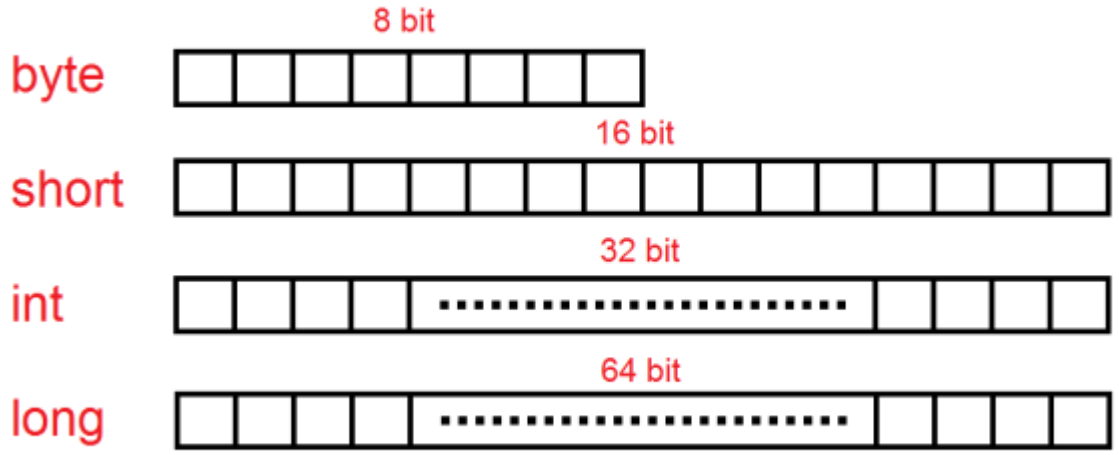
- İşaretsiz (tamamı pozitif sayı olarak kabul edilir, işaret biti kullanılmaz)

- İşaretsiz (1-bit işaret biti olarak ayrılır, 2^{n-1} bit sayılar için kullanılır)



Diğer Tamsayı Veri Tipleri

- Java dilinde aşağıdaki tamsayı türleri mevcuttur



Reel sayılar

- Kayan noktalı reel sayılar
- Sabit noktalı reel sayılar

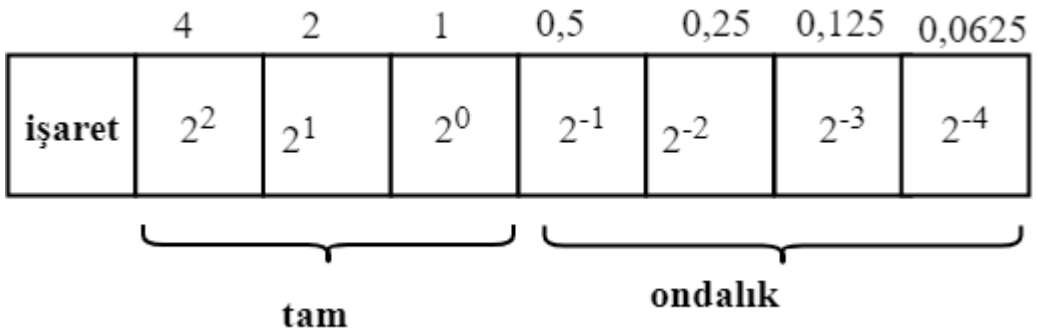
Olmak üzere ikiye ayrılmaktadır.

0,23 $23x10^{-2}$

Reel Sayılar

- Sabit noktalı reel sayılar
- 3 alandan oluşur: işaret, tam, ondalık

- Örnek: 3.1875



0011 (tam) =3

0011 (ondalık) =0.1875

Sayı=tam+ondalık

Tam gösterimi: 00110011

(1-işaret biti, 3-bit tam, 4-bit ondalık olarak varsaydık)

Sabit Noktalı Reel Sayılar

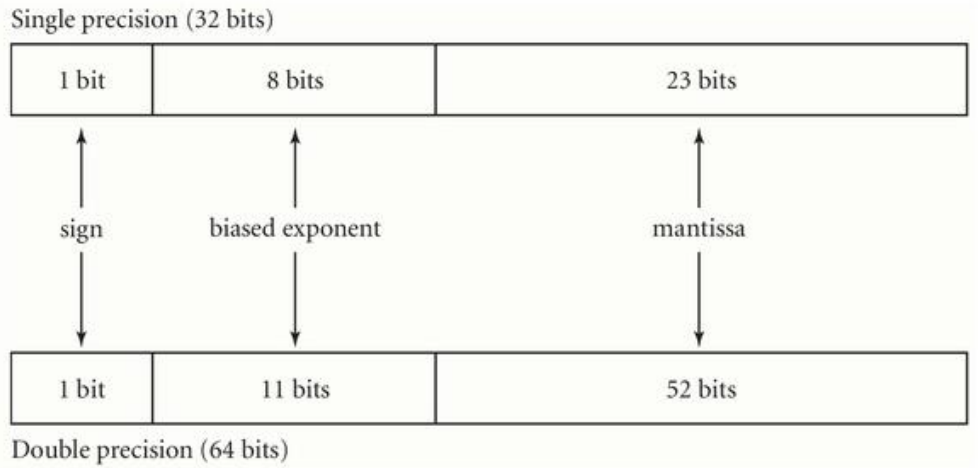
- Örnek: 6.75 sayısı

İşaret	4	2	1	0.5	0.25	0.125	0.0625
0	1	1	0	1	1	0	0

= 6.75

Kayan Noktalı Reel Sayılar

- Tek Duyargalı Reel Sayılar
- Çift Duyargalı Reel Sayılar



https://www.oreilly.com/library/view/java-illuminated-4th/9781284045314/Images/FM_page_1197_1.jpg

- Terimler:

Sign: İşaret biti

Bias/Exponent: Üs

Mantissa: Anlamlı kısım

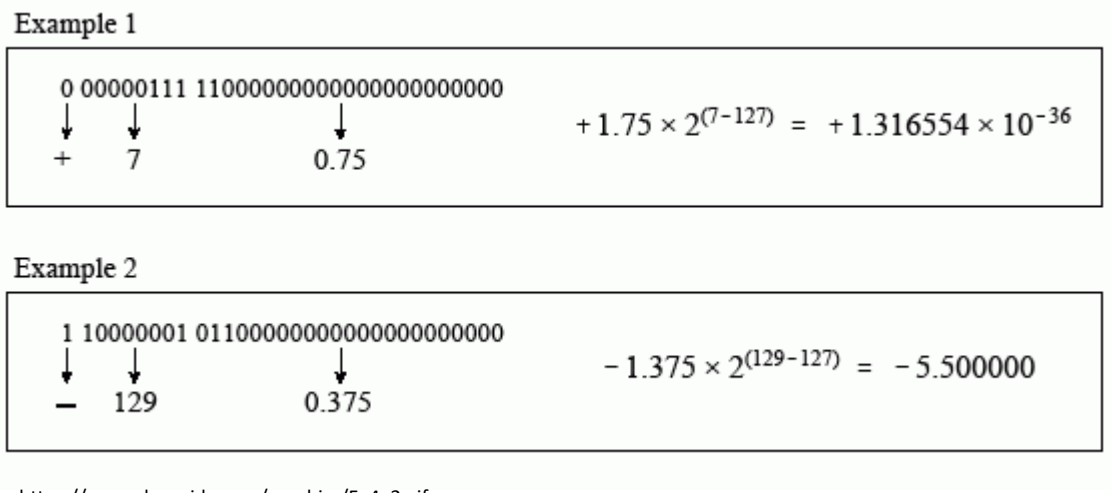
Sayı=işaret*$(1+M*2^{(E-127)})$ dönüşümü ile ondalık karşılık bulunur.

Kayan Noktalı Reel Sayılar

Örnek: kayan noktalı tipten ondalık sayışa dönüşüm

Sayı=işaret*(1+M*$2^{(E-127)}$)

E değeri 127'den büyük ise üs pozitiftir!



https://www.dspguide.com/graphics/F_4_2.gif

Kayan Noktalı Reel Sayılar

Örnek: Ondalık sayıdan kayan noktalı tipe dönüşüm (6,375)

$6 \rightarrow (110)_2$

$0{,}375 = (0{,}011)_2 \rightarrow 6{,}375 = (110{,}011)_2$

Sayıyı olağan duruma getirirsek	: $110{,}011 = 1{,}10011 \times 2^2$
Sayı > 0 olduğundan işaret biti	: 0
Sayının üst değerinin saptırılmış hali	: $2+127 = 129 \rightarrow 129_{10} = 10000001_2$
Anlamlı kısım	: 10011000000000000000000
Sayı son olarak	: 0 10000001 10011000000000000000000
şeklinde ifade edilir.	

http://yzgrafik.ege.edu.tr/~ugur/12_13_Fall/DS/DataStructures_Part_4.pdf

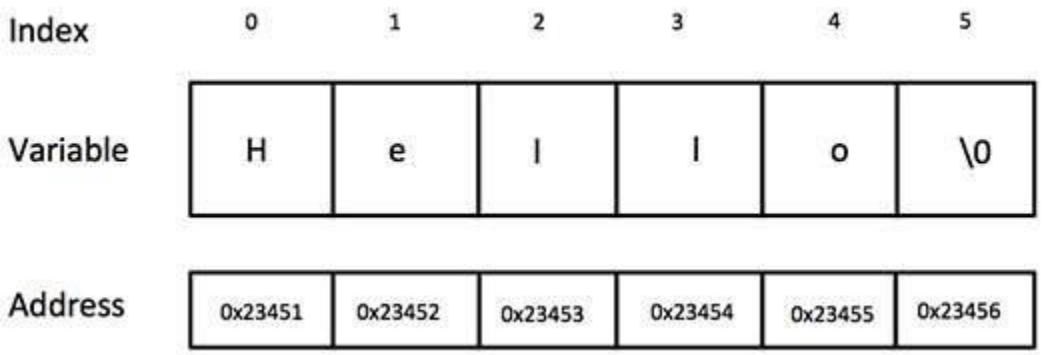
Pratik Yapmak İçin

- İşaretli/işaretsiz tam sayı dönüşümleri
- Sabit/kayan noktalı reel sayılar için
- vb

Google ortamında onlarca online tool bulunmakta

Sözcükler

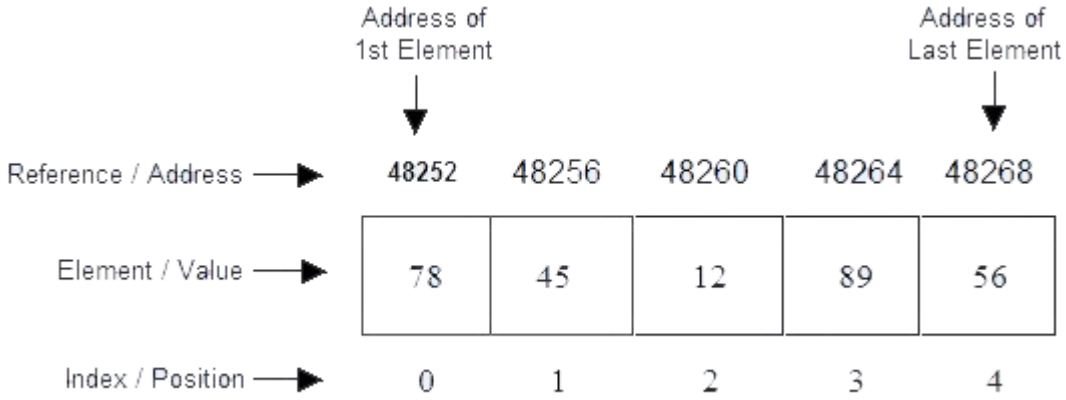
- Sözcükteki her bir karakter için bir alan (ascii/unicode) ve bir sonlandırma karakteri
- Özel karakterler kaçış karakteri ile birlikte kullanılırlar.(\)



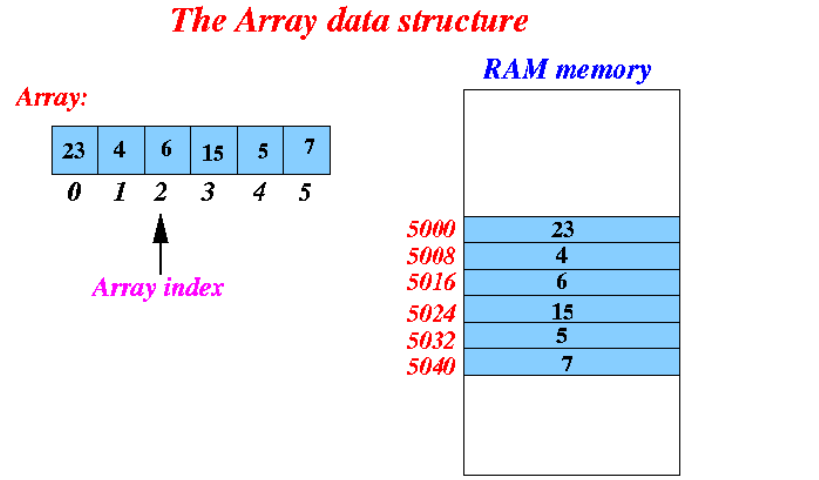
https://www.tutorialspoint.com/cplusplus/images/string_representation.jpg

Diziler

- Diziler aynı tipte ardışık elemanlar içerirler. Çoğunlukla dizi tanımlanırken bellekte yer ayrılması için boyutu belirtilmelidir(Python gibi diller istisna).



http://www.tutorialdost.com/Java-Programming-Tutorial/images/Java-Array.png



http://www.mathcs.emory.edu/~cheung/Courses/171/Syllabus/1-intro/FIGS/170/array02x.gif

Diğer Veri Türleri

- Temel tipler çoğu dil için ortaktır
  - Karakter, tamsayı, reel sayı, sözcük, dizi
- Python tarafından desteklenen bazı veri türleri
  - Liste, tüp, sözlük
- Diğer veri türleri
  - Struct, Union, Nesne

Veri Modelleri

- Bağlı Liste
- Ağaç
- Graf
- Veri tabanı
- Durum makinesi
- Ağ

Sezgisel Algoritmalar

- Karmaşık problemlerin çözümünde sezgisel algoritmalar kullanılmaktadır. Sezgisel algoritmalar her zaman aynı sonucu/aynı doğruluğu elde edemeyebilir. Elde edilecek sonucun doğruluğunun ispatlanması mümkün olmayabilir.

  - A*
  - Arı sürüsü algoritması
  - Genetik algoritmalar
  - Vb…

Kaba kod/Sahte kod

- Algoritmanın temsili yada anlaşılması için (Bir programlama dilinde kodlamadan önce) birçok yol vardır.
  - Kaba kod(pseduo code): Konuşma diline yakındır.
  - Akış diyagramları: Programlama dilindeki yapılar evrensel şekiller ile ifade edilir.
  - N-S/W-O diyagramları: akış diyagramlarının alternatifleridir.

Bir Sonraki Ders

- Algoritma analizi
  - Algoritma kompleksliği (Big-O) notasyonu
  - Algoritma hızı/Bellek Gereksinimi
  - En iyi/En kötü/Ortalama durum
  - Benchmark