YZM 3207- ALGORİTMA ANALİZİ VE TASARIM

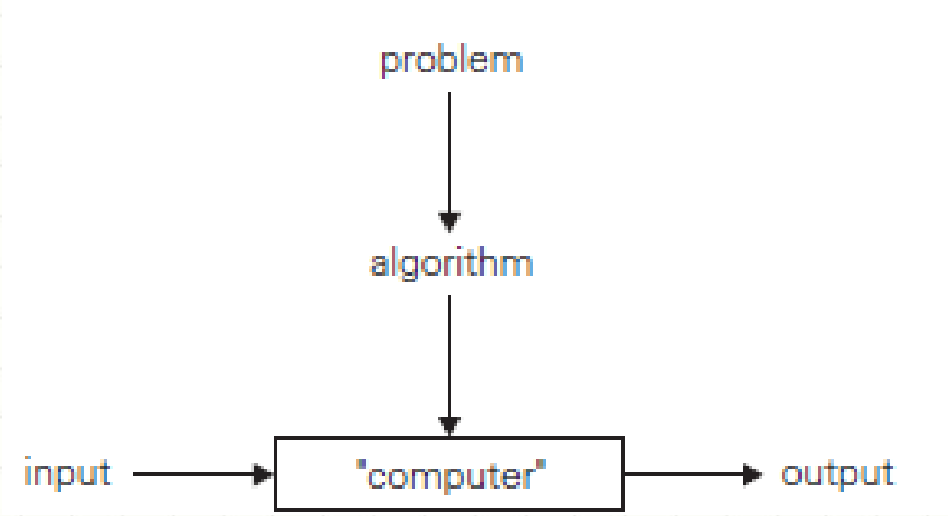
DERS#1: ALGORİTMA KAVRAMI

Algoritma Nedir?

- Algoritma
  - Bir problemin çözümü için geliştirilmiş özel metot
  - Girdileri çıktılara dönüştüren sıralı hesaplama adımları
  - Tanımlanmış bir problemin çözümü için kullanılan araç

  - *«Bir problemin çözümü için izlenilen sıralı ve anlaşılır buyruklar»*

Algoritma Hedefi



- Algoritma ile hedeflenen
  - Sonlu bir zaman içinde
  - Belirli girdiler ile
  - İstenilen çıktıyı elde etmek

Nasıl bir algoritma?

- Bir algoritma
  - Birden fazla biçimde sunulabilmeli
  - Net ve anlaşılır olmalıdır
  - Etkin ve faydalı olmalıdır
  - Sonlu veya sonlandırılabilir olmalıdır
  - Geliştirildiği problem için doğru olmalıdır

Algoritmanın Tarihçesi

- El Harezmi
  - 9. yyda yaşamış bir matematikçi
  - Algoritma ve cebir kavramlarının «babası» olarak bilinir
  - 0 sayısını ve x bilinmeyenini ilk kullanan kişi
- Euclid
  - En büyük ortak böleni bulma problemi için geliştirdiği çözüm ilk algoritmalardan biri olarak kabul ediliyor

En Büyük Ortak Böleni Bulma Problemi

- EBOB (Greater Common Divisor - gcd) bulma problemi
  - İki negatif olmayan tam sayıyı kalansız bölen en büyük sayının bulunması
  - gcd(m, n) = ?
- Euclid Çözümü
  - (m mod n) işleminin sonucu 0 olana kadar
    - gcd*(m, n)* = gcd*(n, m* mod *n) işlemini tekrar et*
      - gcd*(*60*,* 24*)* = gcd*(*24*,* 12*)* = gcd*(*12*,* 0*)* = 12.

EBOB Problemi

- Euclid Çözümü
  - (m mod n) işleminin sonucu 0 olana kadar
    - gcd*(m, n)* = gcd*(n, m* mod *n) işlemini tekrar et*
      - gcd*(*60*, 24)* = gcd*(*24*, 12)* = gcd*(*12*, 0)* = 12.
  - Adım 1 : Eğer *n* = 0 ise m değerini sonuç olarak döndür ve dur
    - Değil ise 2. adıma git
  - Adım2 : (m / n) işlemini yap, kalanı r'ye ata.
  - Adım 3 : n değerini m'ye ata, r değerini n'ye ata (m = n, n = r). Adım 1' dön.
- **Sözde Kod**

```
//Computes gcd(m, n) by Euclid's algorithm
//Input: Two nonnegative, not-both-zero integers m and n
//Output: Greatest common divisor of m and n
while n ≠ 0 do
    r ← m mod n
    m ← n
    n ← r
return m
```

EBOB Problemi

- Euclid'in algoritmasının sonu var mı?
  - N değeri her iterasyonda küçülüyor.
  - 0'dan daha küçük, negatif, olamayacağı biliniyor.
    - İki pozitif sayının bölme işleminden negatif sayı veya kalan çıkamaz.
  - Er yada geç sıfıra ulaşarak algoritma duracaktır.

EBOB Problemi

- Ardışık Tamsayı Kontrol Algoritması ile Çözüm (Consecutive integer checking algorithm)
  - EBOB (m,n) ikilisinin küçük olanından daha büyük olamaz
    - Min(m,n) değerinin EBOB olup olmadığı kontrol edilir
      - Min(m,n) EBOB ise işlem sonlanır,
        - » Değil ise bir azaltılarak tekrar kontrol edilir (min(m,n)--)
  - Algoritma
    - **Adım1:** min{*m, n*} değerini *t'ye ata.*
    - **Adım 2:** (*m / t) işlemini yap, Eğer kalan* = 0 ise Adım 3'e git;
      - Değil ise adım 4'e git
    - **Adım 3 :** (n/t) işlemini yap. Eğer kalan = 0 ise *t değerini sonuç olarak döndür*
      - *Değil ise adım 4'e git*
    - **Adım 4:** t değerini 1 azalt, Adım 2'ye git

EBOB Problemi

- Euclid'in algoritmasından farklı olarak bu algoritma
  - Değerlerden biri 0 ise doğru çalışmaz (Neden?)
  - Algoritma girdilerinin kesin ve dikkatli bir şekilde belirlenmesi gerekliliği ve önemi

EBOB Problemi

- Asal Çarpanlara Ayırma
  - EBOB'u bulunacak sayılar asal çarpanlarına ayrılır
    - Ortak asal çarpanların çarpımı EBOB'u verir
  - Adım 1 : m'nin asal çarpanlarını bul
  - Adım 2: n'nin asal çarpanlarını bul
  - Adım 3: Adım1 ve Adım2'de hesaplanan asal çarpanlardan ortak olanları belirle
  - Adım 4: Ortak asal çarpanları çarp

EBOB Problemi

- Asal çarpanlara ayırma algoritmanın işlem karmaşıklığı Euclid'in algoritmasından daha fazla
- Asal çarpanlara ayırma işleminin algoritması açık değil

- Eratosthenes'in Eleme Algoritması?

Asal Sayıları Bulma

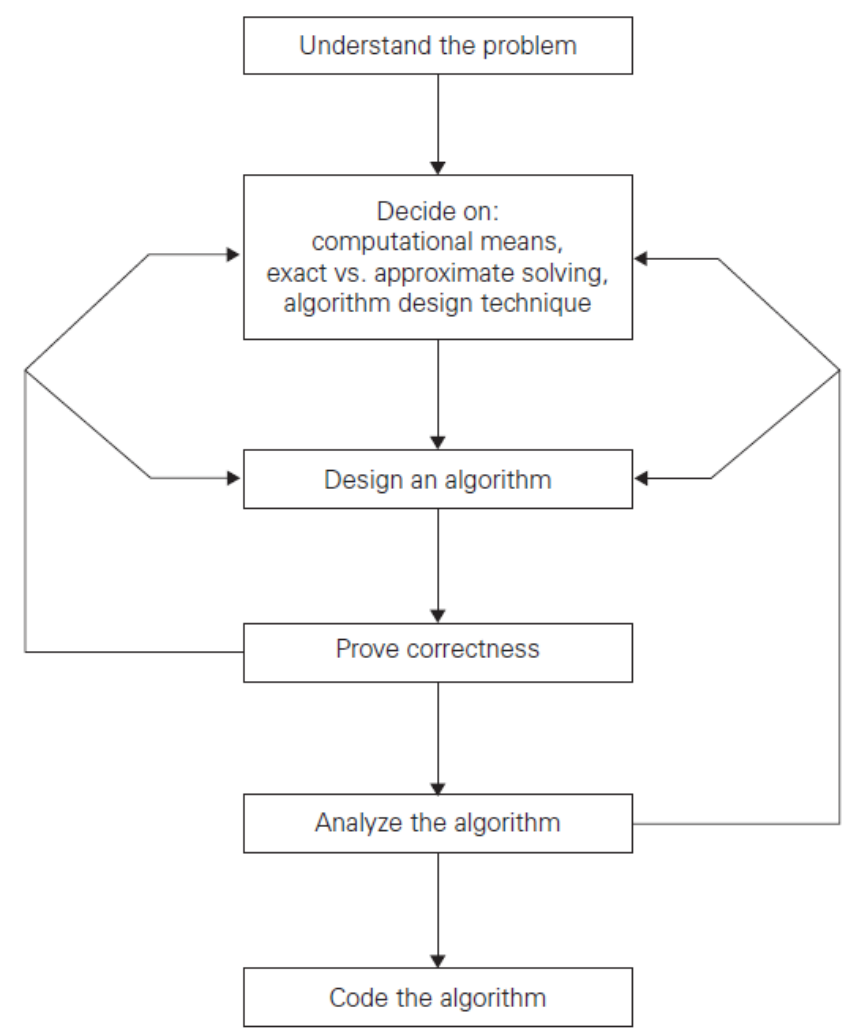
- Eratosthenes'in Eleme Algoritması?
  - 2 ile n arasındaki asal sayıları bulmak için kullanılır
  - 1. iterasyonda 2'nin katları
  - 2. iterasyonda 3'ün katları
  - 3. iterasyonda 5'in katları elenir (?)

2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
2	3		5		7		9		11		13		15		17		19		21		23		25
2	**3**		5		7				11		13				17		19				23		25
2	3		**5**		7				11		13				17		19				23		

Asal Sayıları Bulma

```
for p ← 2 to n do A[p] ← p
for p ← 2 to ⌊√n⌋ do      //see note before pseudocode
    if A[p] ≠ 0           //p hasn't been eliminated on previous passes
        j ← p * p
        while j ≤ n do
            A[j] ← 0      //mark element as eliminated
            j ← j + p
//copy the remaining elements of A to array L of the primes
i ← 0
for p ← 2 to n do
    if A[p] ≠ 0
        L[i] ← A[p]
        i ← i + 1
return L
```

Algoritma Tasarım ve Analiz Süreci



Algoritma Tasarım ve Analiz Süreci

- Problemin Anlaşılması
  - Problem net bir şekilde ifade edilmeli
    - Anlaşılmayan hususlar için sorular
    - Gerekliyse elle işletme
    - İstisnai durumlar?
- Kullanılacak Donanımın Yetenekleri
  - RAM model
    - Birçok temel algoritmanın denendiği sistem

Algoritma Tasarım ve Analiz Süreci

- Kesin Sonuç – Yaklaşık Sonuç Seçimi
  - Tam sonuca ulaşılabilir mi? (Exact Algorithm)
  - Yaklaşık sonuç mu bulunacak? (Approximation Algorithm)
    - Lineer olmayan denklemler
    - Karekök
  - Bazı durumlarda kesin sonuca ulaşmak için
    - Uzun işlem süresi
    - Karmaşık işlemler gerekiyorsa yaklaşık sonuç kabul edilebilir

Algoritma Tasarım ve Analiz Süreci

- Algoritma Tasarım Tekniğinin Belirlenmesi
  - Algoritma Tasarım Tekniği
    - Çözüm için kullanılabilecek algoritmik bir yaklaşım
    - Çeşitli hesaplama problemlerinde uygulanabilir olmalı
- Algoritma Tasarımı ve Veri Yapıları
  - Bellek kullanımı
  - Veri türü
  - Donanım?

Algoritma Tasarım ve Analiz Süreci

- Algoritma Açıklama Yöntemleri
  - Akış diyagramı
  - Sözde kod
  - Kelimeler ile anlatım
- Algoritmanın doğruluğunun kanıtlanması
  - Denemeler
  - İstisnalar
  - İspatlar

Algoritma Tasarım ve Analiz Süreci

- Algoritmanın Analizi
  - Algoritmanın doğruluğu en önemli unsur
  - Etkinlik
  - Zaman Etkinliği
  - Basitlik
- Algoritmanın kodlanması
  - Belirlenen özelliklerin, veri yapılarının uygulanabileceği bir dil seçilmesi

Önemli Problem Türleri

- Sıralama
- Arama
- String İşlemleri
- Graph problemleri
- Kombinasyonel Problemler
- Geometrik problemler
- Nümerik problemler

Önemli Problem Türleri

- Sıralama
  - Bir listedeki öğeleri artan sırada düzenlemek
    - Girdi : n adet sayıdan oluşan dizi $<a_1, a_2, \ldots, a_n>$
    - Çıktı : Girdinin $a'_1 \leq a'_2 \leq \ldots \leq a'_n$ *Şeklinde yeniden düzenlenmesi*
  - Neden Sıralama?
    - Arama işlemini kolaylaştırmak
    - Birçok algoritmanın altyordamı
  - Sıralama Anahtarı
    - Veri bütününün sıralamayı yönlendirmek için seçilmiş özel parçası
      - Bir başarı listesinin sıralanması (İsme, numaraya, nota göre)

Önemli Problem Türleri

- Örnek sıralama algoritmaları
  - Selection
  - Bubble sort
  - Insertion sort
  - Merge sort
  - Heap sort …
- Sıralama algoritması karmaşıklığını değerlendirmek
  - Yapılan Karşılaştırma sayısı
- Sıralama algoritmaları için iki önemli özellik
  - Stabilite: İki eşit elemanın birbirlerine göre sıralanmadan önceki pozisyonlarında kalması
    - Bir dizide birbirine eşit iki eleman var.
    - Sıralamadan önceki konumları i ve j, i<j
    - Sıralamadan sonraki konumları i’ ve j’ ve i’<j’ ise algoritma stabil bir sıralama algoritmasıdır
  - Yerinde (In place) :
    - Fazladan bellek alanı gerektirmeyen sıralama algortimaları
    - İstisna olarak küçük bellek birimleri kullanabilir
      - Dizinin tamamı kadar değil, bir değişken kadar

Önemli Problem Türleri

- Arama
  - Bir değeri verilen veri seti içinde bulma
- Sıralama algoritması örnekleri
  - Sıralı arama
  - İkili arama…
- Her durum için ideal arama algoritması yok
  - Hızlı fakat fazla bellek alanı
  - Sadece sıralı verilerde arama

Önemli Problem Türleri

```
Procedure ikiliarama(x:integer, a1,a2,...,an: artan tamsayılar)
i:=1; {i, arama aralığının sol bitiş noktasını gösterir}
j:=n; {j, arama aralığının sağ bitiş noktasını gösterir}
while i<j do
begin
   m:=[(i+j)/2];
   if x>am then i:=m+1;
   else j := m;
end
if x=ai then konum:=i
else konum:=0;
{konum, x'e eşit olan terimin indisidir veya eğer x
   bulunamamış ise değeri sıfırdır}
```

Önemli Problem Türleri

- Sıralı dizi içerisinde 19 değeri aranıyor
- 1,2,3,5,6,7,8,10,12,13,15,16,18,19,20,22
  - 1,2,3,5,6,7,8,10
  - 12,13,15,16,18,19,20,22
    - 12,13,15,16
    - 18,19,20,22
      - 18,19
      - 20,22

Önemli Problem Türleri

- String İşleme
  - String
    - Bir alfabede yer alan karakterlerden oluşmuş dizi
  - Alfabetik, nümerik, özel karakterlerden oluşabilir
  - Örnek
    - Bir metin içerisinde www ifadesini aramak
    - @ karakterini aramak
    - Derleyiciler

Önemli Problem Türleri

- Graph Problemleri
  - Graph
    - Birbirine sınır adı verilen hatlarla bağlı noktalar grubu
  - Gerçek hayat problemleri
    - WWW modellemesi
    - Haberleşme ağları
    - Proje planlaması
  - Graph örnekleri
    - En kısa yol
    - Topolojik sıralama

Önemli Problem Türleri

- Kombinasyonel problemler
  - Çeşitli kısıtlılıkları sağlayan kombinasyonel çözümlerin bulunması
    - Maksimum değer, minimum maliyet
- Sorunlar
  - Probleme göre kombinasyonel çözümler çok büyüyebilir
  - Bu tip problemleri, kabul edilebilir bir sürede çözebilecek bilinen bir algoritma yok

Önemli Problem Türleri

- Geometrik problemler
  - Nokta, doğru, çokgen gibi geometrik nesneler ile ilgilenen formülleri
- Uygulama Alanları
  - Robotik
  - Bilgisayar Grafikleri
  - Tomografi
- En bilinen problemler
  - En yakın çift (Closest Pair)
  - Dışbükey gövde (Convex Hull)

Önemli Problem Türleri

- Nümerik Problemler
  - Yaklaşık çözüme ulaşılmış problemlerin kesin sonuca ulaştırılmaya çalışılması
    - Denklem, denklem sistemi çözümleri
    - Kısıtlı integraller

En Yakın Çift Problemi

Verilen nokta kümesi içerisinde birbirine en yakın noktaları bulmak

NOKTA	X KOORDİNATI	Y KOORDİNATI
A	3	2
B	2	5
C	5	-3
D	-2	0
E	4	6
F	0	-4

İki Nokta Arası Mesafe : $\sqrt{(X1 \quad -X2 \quad)\text{^}2 + (Y1 \quad -Y2 \quad)\text{^}2}$

Metin Arama

- Bir metin içerisinde aranan ifadeyi bulmak
  - Büyük / Küçük Harf duyarlılığı ?
  - Tam uyum / Ek almış ?