PERFORMANS GÖREVİ

Cebirsel ve Fonksiyonel İşlemleri Algoritmik Dille Yapılandırarak Akış Şeması Oluşturma

---

Adı Soyadı: .....

Sınıf / Şube: .....

Numara: .....

Teslim Tarihi: .....

Beklenen Performans:

Verilen iki farklı işlemi ayrı ayrı algoritmik dille yapılandırarak bu işlemlerin akış şemalarını oluşturma ile ilgili sunum/rapor hazırlama.

Görev:

Bu görevde aşağıdaki iki işlem ayrı ayrı algoritmik dille ifade edilmiş ve her biri için akış şeması oluşturulmuştur:

- Herhangi iki doğal sayının EBOB (Ebob) ve EKOK (Ekok) değerlerini bulan algoritma,
- İki kenar uzunluğu ve bu iki kenar arasındaki açı ölçüsü verilen bir üçgenin alanını bulan algoritma.

1. Çalışma Planı

Göreve başlamadan önce aşağıdaki çalışma planı oluşturulmuş; yapılacak işler ve tahmini çalışma takvimi belirlenmiştir:

Aşama	Yapılacak İş	Süre (Tahmini)
1	Görevin ve beklenen performansın incelenmesi, kaynak taraması	1. Gün
2	EBOB-EKOK algoritmasının adımlarının belirlenmesi (algoritmik dil ile yazımı)	2. Gün
3	EBOB-EKOK akış şemasının çizilmesi ve kontrol edilmesi	3. Gün
4	Üçgen alanı algoritmasının adımlarının belirlenmesi (algoritmik dil ile yazımı)	4. Gün
5	Üçgen alanı akış şemasının çizilmesi ve kontrol edilmesi	5. Gün
6	Her iki algoritmanın örnek değerlerle elle test edilmesi	6. Gün
7	Sunumun/raporun düzenlenmesi, gözden geçirilmesi ve teslime hazırlanması	7. Gün

2. Çalışmada Takip Edilecek Adımlar

Her iki algoritma için de aşağıdaki genel yöntem izlenmiştir:

- Problem durumu okunarak girdiler (verilen değerler) ve çıktılar (istenen sonuçlar) belirlenir.
- Problemin çözümü için gereken matematiksel işlemler / formüller ifade edilir.
- Çözüm süreci, sıralı adımlar hâlinde algoritmik (doğal) dille yazılır.
- Gerekiyorsa algoritmaya karar (koşul) ve/veya tekrar (döngü) yapıları eklenir.
- Algoritmik dille ifade edilen adımlar, standart akış şeması sembolleri kullanılarak görselleştirilir (Başla/Bitir, Giriş/Çıkış, İşlem, Karar).
- Örnek değerlerle algoritma elle işletilerek doğruluğu (test edilerek) kontrol edilir.

3. Birinci İşlem: İki Doğal Sayının EBOB ve EKOK'unu Bulma

3.1. Problemin Tanımı

Girdi olarak iki doğal sayı (a, b) verilmektedir. Bu iki sayının:

- En Büyük Ortak Bölen'i (EBOB),
- En Küçük Ortak Katı'nı (EKOK)

bulan bir algoritma oluşturulacaktır. EBOB, Öklid (Euclid) Algoritması kullanılarak; EKOK ise EBOB bulunduktan sonra

$$\mathbf{EKOK(a, b) = (a \times b) / EBOB(a, b)}$$

bağıntısı ile elde edilecektir. Öklid Algoritması'nın mantığı: iki sayıdan büyüğü küçüğüne bölünür, kalan bulunur; kalan sıfır oluncaya kadar (küçük sayı) ile (kalan) aynı işleme tabi tutulur. Kalan sıfır olduğunda son bölen, aranan EBOB değeridir.

3.2. Algoritmik Dil ile İfade

1. Adım: Başla.

2. Adım: Girdilerin alınması: a ve b doğal sayılarını oku.

3. Adım: Değişkenlerin tanımlanması: x = a, y = b olarak ata (a ve b'nin özgün değerleri EKOK hesabı için saklanır).

4. Adım: Döngü / kontrol: Eğer y ≠ 0 ise; kalan = x mod y, x = y, y = kalan olarak güncelle ve bu adımı tekrarla. Eğer y = 0 ise 5. adıma geç.

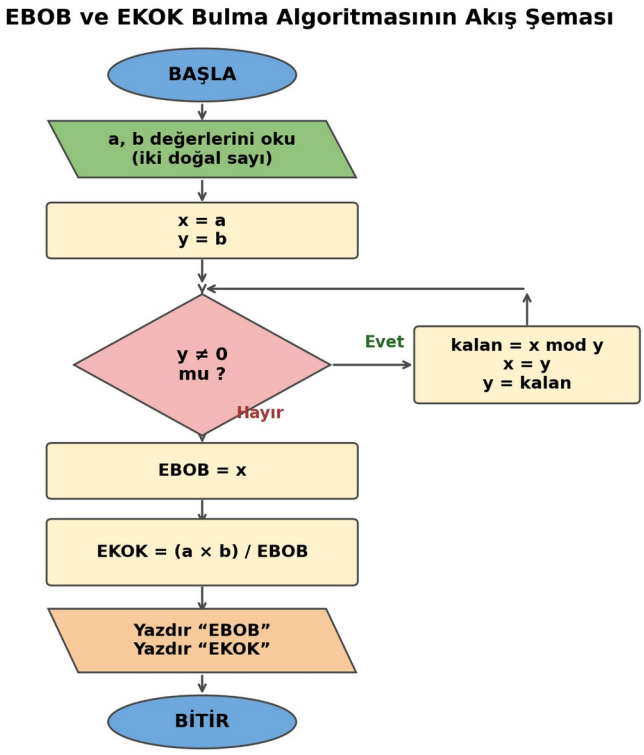
5. Adım: EBOB'un belirlenmesi: EBOB = x (döngü sonunda x'in aldığı değer).

6. Adım: EKOK'un hesaplanması: EKOK = (a × b) / EBOB.

7. Adım: Sonuçların yazdırılması: "EBOB" ve "EKOK" değerlerini yazdır.

8. Adım: Bitir.

3.3. Akış Şeması



Görsel 1: İki doğal sayının EBOB ve EKOK'unu bulan algoritmanın akış şeması

3.4. Örnek Kontrol (Elle Test)

a = 18, b = 24 için: x=18, y=24 → x=24,y=18 → kalan=24 mod18=6, x=18,y=6 → kalan=18 mod6=0, x=6,y=0 → y=0 olduğundan EBOB=6. EKOK = (18×24)/6 = 432/6 = 72. Gerçekten de 18 ve 24'ün EBOB'u 6, EKOK'u 72'dir; algoritma doğru sonucu vermektedir.

4. İkinci İşlem: İki Kenarı ve Aralarındaki Açısı Verilen Üçgenin Alanını Bulma

4.1. Problemin Tanımı

Bir üçgenin iki kenar uzunluğu (a, b) ve bu iki kenar arasındaki açının ölçüsü (C, derece cinsinden) verildiğinde üçgenin alanını hesaplayan bir algoritma oluşturulacaktır. Trigonometrik alan bağıntısı:

$$\textbf{Alan} = (1 / 2) \times a \times b \times \sin(C)$$

kullanılacaktır. Programlama dillerindeki/hesap makinelerindeki çoğu sin fonksiyonu açıyı radyan cinsinden kabul ettiğinden, hesaplamadan önce C açısı derece → radyan dönüşümüne tabi tutulur: C_radyan = C × (π / 180). Ayrıca girilen değerlerin geçerliliğini (a>0, b>0 ve 0°<C<180°) kontrol eden bir karar adımı algoritmaya eklenmiştir.

4.2. Algoritmik Dil ile İfade

1. Adım: Başla.

2. Adım: Girdilerin alınması: a, b kenar uzunluklarını ve C açısını (derece) oku.

3. Adım: Kontrolün yapılması: Eğer a>0 ve b>0 ve 0<C<180 ise 4. adıma geç; değilse "Geçersiz veri" yazdır ve 7. adıma geç.

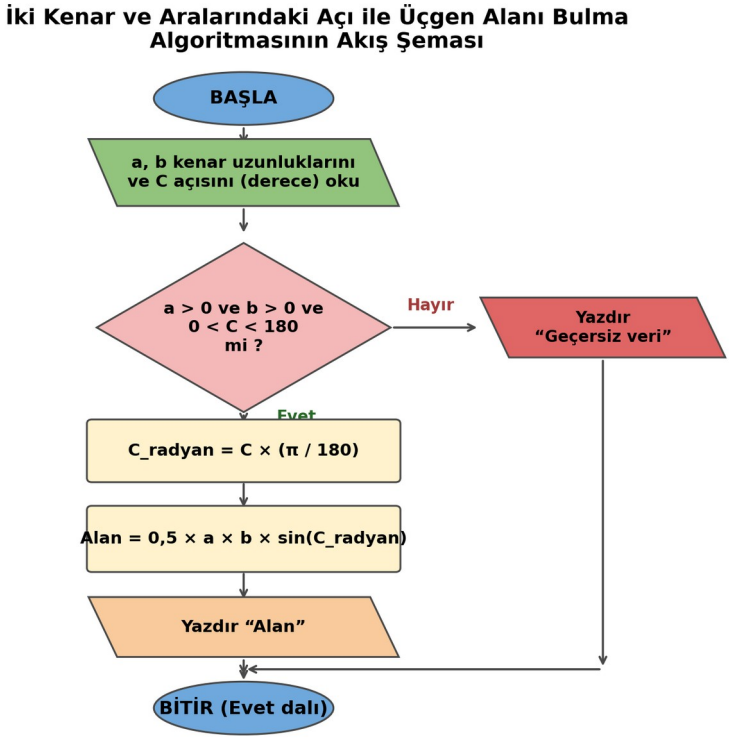
4. Adım: Birim dönüşümü: C_radyan = C × (π / 180) hesapla.

5. Adım: Alanın hesaplanması: Alan = 0,5 × a × b × sin(C_radyan) hesapla.

6. Adım: Sonucun yazdırılması: "Alan" değerini yazdır.

7. Adım: Bitir.

4.3. Akış Şeması



Görsel 2: İki kenar ve aralarındaki açı ile üçgen alanı bulan algoritmanın akış şeması

4.4. Örnek Kontrol (Elle Test)

a = 6 cm, b = 8 cm, C = 30° için: Doğrulama koşulu sağlanır (a>0, b>0, 0<30<180). C_radyan = 30×(π/180) ≈ 0,5236. sin(30°) = 0,5. Alan = 0,5 × 6 × 8 × 0,5 = 12 cm². Algoritma beklenen sonucu doğru biçimde vermektedir.

5. Sonuç ve Değerlendirme

Bu çalışmada iki farklı matematiksel problem — iki doğal sayının EBOB ve EKOK'unun bulunması ile iki kenarı ve aralarındaki açısı verilen üçgenin alanının hesaplanması — önce algoritmik (doğal) dille ifade edilmiş, ardından standart akış şeması sembolleri kullanılarak görselleştirilmiştir.

Birinci algoritmada bir tekrar (döngü) ve karar yapısı (Öklid Algoritması'nın kalan bulma süreci) kullanılırken, ikinci algoritmada girdilerin geçerliliğini denetleyen bir karar (koşul) yapısına yer verilmiştir. Her iki algoritma da örnek sayısal değerlerle elle test edilmiş ve beklenen matematiksel sonuçları doğru şekilde ürettiği görülmüştür.

Bu çalışma, algoritmik düşüncenin farklı problem türlerine (sayı teorisi ve geometri) nasıl uygulanabileceğini; aynı yapının (Başla → Girdi → İşlem/Karar → Çıktı → Bitir) çok çeşitli problemlerin sistematik çözümünde kullanılabileceğini göstermektedir.