

NYP-3. Hafta Laboratuvar Ödev Soruları

- 1) Kullanıcıdan bir dizi tam sayı alın ve bu sayıları sıralayın. Ardından, kullanıcıdan bir sayı alın ve bu sayının dizide olup olmadığını ikili arama algoritması ile kontrol edin. Sonucu ekrana yazdırın.
- 2) Kullanıcıdan pozitif tam sayılar alarak, bu sayıların ortalamasını ve medyanını hesaplayan bir program yazın. Kullanıcı 0 girene kadar sayıları almaya devam etsin. 0 girildiğinde ortalamayı ve medyanı gösterin.
- 3) Kullanıcıdan bir dizi tamsayı alın ve bu dizideki ardışık sayı gruplarını tespit eden bir program yazın. Örneğin, 1, 2, 3, 5, 6, 7, 10 dizisi için program, 1-3 ve 5-7 gruplarını döndürmelidir. Kullanıcı 0 girene kadar sayıları almaya devam etsin.
- 4) Kullanıcının girdiği matematiksel ifadeyi (örneğin, $3 + 4 * 2 / (1 - 5) ^ 2 ^ 3$) işlem önceliklerine göre çözümleyen bir program yazın. Program, sonucu yazdırmadan önce ifadenin çözüm sürecini açıklamalıdır (hangi işlemlerin hangi sırayla yapıldığını gösterin).
- 5) Kullanıcıdan iki polinom girmesini isteyin (örneğin, $2x^2 + 3x - 5$ ve $x^2 - 4$). Program, bu iki polinomu toplayıp ve çıkararak sonuçları göstermelidir. Kullanıcı, exit yazarak işlemi sonlandırana kadar yeni polinomlar girmeye devam etsin.
- 6) Bir grup bilim insanı, insanları zamanda geriye götürebilecek bir cihaz geliştirdi. Ancak bu cihazı kullanmak için doğru tarih ve saat koordinatlarını çözmek gerekiyor. Cihazın çalışma mantığına göre, tarihin gün, ay ve yıl bileşenleri birbiriyle matematiksel olarak ilişkilendirilmiş durumda. Bir zaman yolcusu, geçmişe gitmek için cihazı kullanmak istiyor, ancak cihazın gideceği tarihi doğru bir şekilde çözmesi gerekiyor. Bilim insanları bu zaman yolcusuna cihazın algoritmasını anlaması için bazı ipuçları verdi:
 - i. **Gün, ay ve yıl ilişkisi:** Cihaz yalnızca belirli tarih formatında çalışıyor. Bir tarih (gün, ay, yıl) cihazın kabul edeceği formattaysa, zamanda yolculuk mümkün oluyor. Kabul edilen formattaki bir tarihin, aşağıdaki koşullara uyması gerekiyor:
 - Gün sayısı asal sayı olmalı.
 - Ay sayısının tüm basamaklarının toplamı çift olmalı.
 - Yıl sayısı ise şu özelliğe sahip olmalı: Yıl sayısını oluşturan rakamlar toplamı, o yılın dörtte birinden küçük olmalı.
 - ii. **Algoritmik zorluklar:** Zaman yolcusu, 2000 ile 3000 yılları arasında bir tarihe gitmek istiyor. Ona cihazın kabul ettiği tüm uygun tarihleri listeleyen bir algoritma yazması gerektiği söyleniyor. Ancak milyonlarca olası tarih kombinasyonu olduğu için, algoritmanın verimli çalışması gerekiyor.
 - iii. **Ek koşullar:** Zaman yolcusunun yalnızca geleceğe gitmesine izin veriliyor. Bu yüzden algoritma, şu andan sonraki bir tarihe odaklanmalı.
 - iv. **Görev:** Zaman yolcusuna yardım etmek için bir algoritma yaz. Bu algoritma, belirlenen tarihler aralığında cihazın kabul edeceği tüm geçerli tarihleri listelemelidir. Her tarih, gün, ay ve yıl formatında olmalı. Cihazın kabul ettiği tarihler listeye eklenmelidir.
- 7) Bir zamanlar efsanevi bir şehir olan Zarva, sayısız hazineyi saklayan devasa bir labirentin içinde kayboldu. Şehre ulaşabilmek için bu labirentin doğru yollarını çözmeniz gerekiyor. Labirent, $M \times N$ boyutlarında bir ızgaradır ve her hücrede bir "kapı" bulunur. Kapılar, sadece belirli matematiksel kuralları karşılayan

koordinatlarda açılabilir. Eğer bir kapı açılabiliyorsa o hücreye geçiş mümkündür, aksi takdirde o hücreye girmek imkansızdır. Şehir haritası şu şekilde tanımlanmıştır:

- i. Labirentteki bir hücreye ancak şu şartlar sağlanırsa girebilirsiniz:
 - Hücresinin x ve y koordinatlarının her iki basamağı da asal sayı olmalıdır.
 - Eğer hem x hem de y koordinatlarının toplamı, x ve y koordinatlarının çarpımına bölünebiliyorsa, kapı açılacaktır.
 - ii. Şehir, labirentin sağ alt köşesinde ($M-1$, $N-1$) konumunda bulunmaktadır. Yolculuğunuz labirentin sol üst köşesinden (0 , 0) başlayacaktır. Eğer şehre ulaşamazsanız, kaybolursunuz.
 - iii. **Görev:** Labirenti çözmek için bir algoritma yazmanız gerekiyor. Bu algoritma, belirlenen koşullara göre hangi hücrelere gidilebileceğini belirleyecek ve başlangıç noktasından hedef noktaya ulaşabilecek bir yol bulup bulamayacağını kontrol edecektir. Eğer bir yol bulabiliyorsanız, yolun adımlarını listeleyin. Eğer hiçbir yol yoksa, "Şehir kayboldu!" çıktısını verin.
- 8) Bir casus örgütü, düşmanlarının mesajlarını çözmelerini zorlaştırmak için son derece karmaşık bir şifreleme sistemi geliştirdi. Bu şifreleme sistemi, mesajların belirli kurallara göre dönüştürüldüğü bir dizi adımdan oluşuyor. Göreviniz, bu sistemi çözerek şifrelenmiş bir mesajın orijinal haline ulaşmaktır. Sistemin çalışma şekli şu kurallara dayanmaktadır:
- i. **Adım 1 - Fibonacci Dönüşümü:** Mesajın her bir karakteri ASCII değerine dönüştürülür. Daha sonra her karakterin ASCII değeri, o karakterin pozisyonuna göre bir Fibonacci sayısı ile çarpılır. Mesajdaki karakterlerin sırası 1'den başlar. Yani ilk karakter, Fibonacci(1), ikinci karakter Fibonacci(2) ile çarpılır ve bu işlem tüm karakterler için devam eder. Fibonacci serisi: 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, ...
 - ii. **Adım 2 - Modüler Çözümleme:** Fibonacci ile çarpılan her bir ASCII değeri, bir mod işlemine tabi tutulur. Mod işlemi şu şekilde çalışır:
 - Eğer karakterin pozisyonu asal bir sayıysa, o karakterin değeri 100'e bölümünden kalan alınır.
 - Eğer pozisyon asal değilse, karakterin değeri 256'ya bölümünden kalan alınır.
 - iii. **Adım 3 - Şifreleme:** Elde edilen her mod sonucu, tekrar bir ASCII karakterine dönüştürülür ve yeni şifreli mesaj oluşturulur.
 - iv. **Görev:** Size şifrelenmiş bir mesaj verilecektir. Bu mesajın çözülmesi ve orijinal haline geri dönülmesi gerekmektedir. Ancak şifreleme işlemi sırasında kullanılan Fibonacci dönüşümünü ve modüler çözümlemeyi tersine çevirmelisiniz. Görev, bu algoritmayı çözerek mesajı orijinaline döndürmektir.
- 9) Bir grup uzay madencisi, zengin maden yataklarına sahip bir asteroide iniş yaptı. Ancak asteroitteki madenlerin dağılımı oldukça düzensiz ve tehlikeli yollar içeriyor. Maden yataklarına ulaşmak için madencilerin dikkatlice bir yol planlaması gerekiyor. Asteroit, $N \times N$ boyutlarında bir ızgara olarak modellenmiş durumda. Her hücre bir koordinata karşılık geliyor ve madenciler bu koordinatlar üzerinden maden toplamak zorunda. Asteroit üzerindeki maden toplama sistemi şu şekilde işliyor:
- i. **Enerji harcama:** Her hücrede belirli bir enerji tüketimi var. Madenciler bir hücreden diğerine geçerken belirli bir enerji harcarlar. Ancak bazı hücreler daha fazla enerji harcatır çünkü bu hücrelerde engeller, çukurlar veya volkanik

aktiviteler olabilir. Enerji tüketimi, her hücrede farklı olup pozitif tam sayılar ile ifade edilmiştir.

- ii. **Kısıtlamalar:** Madenciler bir hücreden sadece **sağa, aşağıya** veya **çapraz sağa aşağıya** (diyagonal) hareket edebilirler. Geriye dönüş yoktur, bu yüzden planlamada bu sınırlamalara uymak zorundadırlar.
- iii. **Hedef:** Madenciler, (0, 0) noktasından başlayarak (N-1, N-1) noktasındaki en değerli madeni çıkartmak istiyorlar. Ancak bunu yaparken harcayacakları toplam enerji miktarını en az seviyede tutmaları gerekiyor.
- iv. **Görev:** Bir algoritma yazarak madencilerin (0, 0) noktasından başlayarak (N-1, N-1) noktasına kadar olan en az enerji harcayan yolu bulmalısınız. Enerji maliyeti bir 2D matriste verilecektir. Bu matrisin her hücresi, o hücrede harcanacak enerji miktarını göstermektedir.

10) Gizemli bir adada, her yıl sadece bir kez açılan efsanevi bir kapı vardır. Bu kapı, adanın merkezindeki bir tapınağa götürür. Ancak tapınağa ulaşmak için, kapının açılacağı en uygun zamanı bulmanız gerekmektedir. Kapı, belirli bir dizilimdeki sayılarla kilitlenmiştir ve sadece bu dizilimi doğru çözenler kapıyı açabilir.

Bu dizilim, bir dizi sayı ve operatörlerden oluşmaktadır, ancak bu diziyi çözmek için belirli kurallar vardır. Diziyi doğru çözmek, sadece operatörlerin doğru sıralamasıyla mümkündür ve bu sıralama, sayılar arasındaki ilişkileri çözmeyi gerektirir. Adadaki en büyük bilge, bu diziyi çözebilecek tek kişi olduğunu söylese de çok fazla kombinasyon ve olasılık olduğu için kesin bir çözüm yoktur.

Görev: Adadaki tapınağın kapısını açmak için belirli bir sayı dizisinin içerisindeki operatörlerin doğru sıralamasını bulmanız gerekmektedir. Ancak operatörlerin her biri belirli bir kurala göre hareket eder:

- i. Her sayı, bir diğer sayıya bağlanmak zorundadır.
- ii. Operatörler sadece toplama (+), çıkarma (-), çarpma (*) veya bölme (/) olabilir.
- iii. Ancak her adımda bir sayı veya operatör eklerken belirli şartlara göre hareket etmelisiniz. Örneğin:
 - Bir operatör eklediğinizde, en az bir sayıyı işlem görmüş hale getirmeniz gerekir.
 - İki operatör ardışık eklenemez.
 - Sonuç her zaman sıfırdan büyük olmalıdır.