

Problem Çözme ve Algoritmalar

Özet

Bu Bölümde;
Genel Problem Çözme Kavramlarını,
Klasik Bulmacaları ve Çözümlerini detaylı biçimde öğreneceksiniz.

Ali GÖKAŞAR
Bilişim Teknolojileri Öğretmeni

2. Problem Çözme ve Algoritmalar

2.1. Problem Çözme Kavramları ve Yaklaşımlar

2.1.1. Programlama Nedir

Bir bilgisayar bilimcisi gibi düşünmek ve programlama ne demektir? Bilgisayar bilimcileri genel olarak matematiksel sembolleri, işlemleri ve formülleri kullanır, mühendisler gibi tasarım yaparak farklı sistemler oluşturur ve bilim insanları gibi deney yaparak teknoloji desteği ile çözüm üretir.

Bir bilgisayar bilimcisi için en önemli beceri **problem çözme becerisidir**. Problem çözme; problemleri formüle edebilme, **farklı** ve **yaratıcı** çözüm yolları önerebilme, çözümü kesin ve doğru biçimde ifade edebilme becerisidir. Programlama aynı zamanda bir **düşünme biçimidir**. Programlama hem problem çözme becerisi hem de bilgi işlemsel düşünme becerisine sahip olmayı gerektirir.

Bilgi işlemsel düşünme; bilgisayar biliminden yararlanarak problem çözme, sistem tasarlama ve insan davranışlarını anlama olarak tanımlanabilir.

Problem çözme süreci;

Problemleri bilgisayar ve başka araçlar yardımı ile çözebilir hale getirme
Mantıklı bir şekilde verileri düzenleme ve çözümleme
Model ve benzetim desteği ile verileri sunma
Algoritmik düşünme çerçevesinde çözümleri otomatikleştirme
Kaynakları verimli bir şekilde kullanarak uygun çözümleri tanımlama, çözümleme ve uygulama
Bulunan çözümü farklı problemlere transfer etme ve geliştirme aşamalarından oluşur.

Bilgisayarlar ile çözümleri otomatik hale getirip problemleri daha etkili çözebilecek ve düşünmenin sınırlarını genişleteceksiniz. Bilgisayar bilimi kavramlarını ve ilkelerini öğrendiğiniz zaman gittikçe değişen teknolojik hayata ve iş yaşamına daha iyi hazırlanabileceksiniz.

Farklı bir bakış açısı ile programlama; bilgisayarın donanımına nasıl davranacağını anlatan, bilgisayara yön veren komutlar ve işlemler bütünüdür.

Bir programlama sisteminin iki bileşeni vardır:

Bilgisayara kurulmuş olan bileşen-programlama ortamı
Programcı tarafından oluşturulan algoritma ve program kodları

Kullandığımız programlama ortamı ile programcı tarafından kullanılacak kelime ve komutları oluşturur, program akışını ve mevcut durumu kontrol edebilir, adım adım işlemleri takip edebilir, oluşturduğumuz işlemleri geliştirerek soyutlaştırabiliriz. Kullandığımız programlama dili ile yapmak istediğimiz işlemleri bilgisayarın anlayacağı biçimde ifade edebilir, işlemleri parçalara bölebilir, parçalardan farklı ve anlamlı bütünlük oluşturabiliriz.

2.1.2. Program Nedir?

Program; yapılacak bir işlemi ya da hesaplamayı gerçekleştirmek için birbirini izleyen komut ya da yönergelerden oluşan yapıdır. İşlemler hem **matematiksel** hem de **mantıksal** olabilir.

Örneğin bir formülün sonucunun hesaplanması ya da bir doküman içerisinde belirli bir metnin aranması gibi. Ayrıntılar programlama dillerine göre farklılaşsa bile belirli komutlar her dilde yer alır.

Girdi: Klavyeden, dosyadan veya başka bir aygıtta veri almaktır.

Çıktı: Ekranda veriyi görüntüleme veya veriyi dosyaya veya başka bir aygıtta göndermedir.

Matematik: Toplama, çarpma gibi bazı temel matematiksel işlemleri gerçekleştirir.

Koşullu Yürütme: Belirli durumları sınamak ve komutları uygun bir sıraya göre çalıştırmaktır.

Tekrarlama: Bazı eylemleri genellikle ufak tefek değişikliklerle yineleme işlemidir.

2.1.3. Hata Ayıklama Nedir?

Programlama, karmaşık bir süreçtir ve programcılar programlamada hata(bug) yapabilir. Programlama hatalarını bulma ve düzeltme işlemine hata ayıklama(debugging) denilir.

2.1.3.1. Söz Dizimsel Hatalar

Söz dizimi, programın yapısı ve bu yapı hakkındaki kurallar demektir. Örneğin Türkçede bir cümle büyük harfle başlamalı ve uygun bir noktalama işaretiyle sona ermelidir. Bu kurallara uymayan cümlelere "Söz dizimi hatası içermektedir." diyebiliriz.

2.1.3.2. Çalışma Zamanı Hataları

Bu hatalar ancak program çalıştırdıktan sonra ortaya çıkar. **Hesaplanması mümkün olmayan işlemler** (sıfıra bölünme) ya da **hiç gerçekleşmeyecek koşulların** ($5 < 3$) yürütülmesi gibi durumlarda ortaya çıkar.

2.1.3.3. Anlam Bilimsel Hatalar

Bu durumda program, genellikle hata vermeden çalışır ancak beklenen sonucu üretemez. Bu yüzden programı satır satır çalıştırarak, farklı adımlardaki çıktıları gözlemleyerek nerede mantık hatası yapıldığını bularak program doğru biçimde çalışana kadar bu hataları ayıklamak gerekir.

Yorucu olmasına rağmen, programlamada bilişsel yoğunluk gerektiren ilginç bir süreçtir. **Hata ayıklama deneysel bir yaklaşımdır.**

2.1.4. Günlük Hayatta Problem Çözme

Günlük hayatımızda problemlerimizi çözmek için yaşantımızı etkileyen pek çok karar veririz. Bu kararlar yalnızca yaşantımızı etkilemekle kalmaz, bazen yaşam kalitemizi ve geleceğimizi bile etkileyebilir.

Örneğin karşılaştığımız problemler, televizyonda hangi kanalı seyretsen gibi basit de olabilir, hangi mesleği seçmeliyim gibi çok önemli de olabilir. Yanlış bir karar verilirse zaman ve kaynaklar boşa gidebilir, bu nedenle nasıl doğru karar verildiğini öğrenmek önemlidir.

Geçici Çözümler Değil, Kalıcı Çözümler Üretmek

Çözüm yöntemlerini öğrenmek veya problem çözebilme yeteneğini geliştirmek yerine sadece formül ezberlemek, geçici çözümler bulmaktan fazlası değildir.

Örneğin; balık tüketmek, karnınızı doyurmak için iyi bir yöntem olabilir ama açlık probleminde kalıcı çözüm değildir.

Çözüm 1: Para kazanıp sürekli balık alabilmek

Çözüm 2: Olta alıp balık tutmayı öğrenmek

En iyi kararı vermek aslında problem çözmektir. **İnsan hayatı aslında bir problem çözme sürecidir.** Genellikle bir problemin birden fazla çözümü vardır, her bir çözüm bir alternatif olarak düşünülebilir.

Problem çözme, amaca ulaşabilmek için alternatifler arasından en uygun yolu belirlemektir. Alternatifler, farklı koşul ve beklentilere göre şekillenir.

Problemler çözülmeye çalışılırken dikkate alınması gereken sınırlılıklar ve koşullar ile uyulması gereken kurallar vardır. Bir problemi yazılım geliştirerek çözerken de çeşitli sınırlılıklar vardır:

Kullandığınız Programlama Dili

Çalıştığınız Ortam (Kişisel Bilgisayar, Tablet)

Performans (İşlemci, Hafıza, Disk, Sunucu)

Bu nedenle programcılar için problem çözme, "bir dizi işlemi, belirtilen sınırlılıklara uygun biçimde gerçekleştirebilen programın yazılması" anlamına gelebilir.

Örnek Sorunlar ve Çözümleri

Kıyafetleri İnternet Mağazalarında Deneyerek Alma Sorunu

Bedenimizin Üç Boyutlu Tarayıcı ile Taranmış kopyasının siteye yüklenerek, beğendiğimiz üç boyutlu kıyafet ölçülerinin denemesi

Kıyafetlerin Renk ve Tasarımlarının Değişmemesi Sorunu

Kıyafetlerimize akıllı telefonlar ile desen tasarımı yapıp bunun, elbisemize aktarılması. Elbisemizin kumaşının bükülebilir ekran özelliğinden olması.

Trafikteki araçların sayısının çok olması.

Araçlar şehir içinde yollardan kaldırılır. Hareketli yollar olabilir. Şeritler trafiğin yoğunluğuna göre hareket edebilir. Yürüyen merdiven gibi yürüyen yollar olabilir. Yolların üzerinde de Ulaşım kabinleri olabilir. Lüks çeşitleri de olabilir. Duraklardan yolcu alır. Yolcusunu aldıktan sonra hareketli şeride geçer

2.1.5. Problem Çözme Süreci

Problem çözme farklı biçimlerde düşünmeyi gerektiren bir eylemdir. Öncelikle klasikleşmiş bazı klasik bulmacaları ve bu bulmacaların çözümlerini inceleyelim.

2.1.5.1. Tilki, Kaz, Mısır Çuvalı

Konuşacağımız ilk klasik problem, beraberindeki nesnelere nehrin karşısına taşınması gereken bir çiftçiyle ilgili bulmacadır. Bu çiftçinin bir tilkiyi, bir kazı ve bir mısır çuvalını nehrin karşısına geçirmesi gerekmektedir. Çiftçinin bu işlemi gerçekleştirmek için küçük bir teknesi var ancak bu teknede çiftçi ile birlikte en fazla bir nesneye daha yer vardır. Ne yazık ki tilki ve kaz ile yalnız kalamaz çünkü kaz mısırı yiyebilir. Aynı şekilde kaz ve mısır çuvalı yalnız bırakılamaz çünkü kaz mısırı yiyebilir. Bu koşullarda çiftçi nehrin karşısına tilki, kaz ve mısırı sorunsuz bir şekilde nasıl geçirebilir?



Tekne ile aynı anda en fazla iki nesne taşınabildiğini biliyoruz. Tilki ve kaz aynı kıyıda yalnız bırakılamayacağı gibi kaz da mısır çuvalıyla aynı kıyıda yalnız bırakılamaz.

Çiftçi her seferinde nesnelere birini alabileceğinden çiftçinin her şeyi kıyıya götürmek için birden fazla gidip gelmesi gerekecektir. İlk gidiş sırasında çiftçi tilkiyi alırsa kaz mısır çuvalıyla yalnız kalır ve kaz mısırı yiyebilir. Aynı şekilde çiftçi ilk gidiş sırasında mısır çuvalını alacak olursa tilki kaz ile yalnız kalacak ve tilki kazı yiyecektir. Bu nedenle çiftçi ilk turda kazı almalıdır. Bununla birlikte bu adımdan sonraki tüm adımlar başarısızlığa uğramış gibi görünür.

Kazın uzakta olması son derece iyi bir çözümdür. Ancak ikinci turda, çiftçi tilki ya da mısırı almalıdır. Bununla birlikte çiftçi neyi alıyor olursa olsun geri kalanlar için yakın kıyıya dönerken kaz uzak kıyıda kalmalıdır. Bu tilkinin ve kazın birlikte bırakılacağı ya da kaz ve mısır çuvalının birlikte bırakılacağı anlamına gelir. Bu durumlardan hiçbiri kabul edilemez çünkü bu durumda problem çözülemez.

Çiftçi daha önce açıklandığı gibi ilk turda kazı almak zorundadır. İkinci turda, çiftçinin tilkiyi aldığını varsayalım. Bununla birlikte tilkiyi kaz ile bırakmak yerine çiftçi kazı yanına alarak yakın kıyıya geri götürür. Tilki uzak kıyıda yalnız kalır. Sonra çiftçi dördüncü turda tilkiyi yakın kıyıda yalnız bırakarak mısır çuvalını uzak kıyıya götürür.

Pek çok kişi yakın veya uzak kıyıda birini geri almayı düşünmez çünkü bu bulmaca zordur. Bazı insanlar, sorunun haksız olduğunu önererek "Geri alabileceği söylememişsiniz." şeklinde tepki verebilir. Oysaki problem ifadesinde aksi yönde bir açıklama bulunmamaktadır.



1

2

3

Çözüm Adımı



4

5



6

7

8

Ögelerden birini yakın kıyıya geri götürme imkânı açıkça belirtilseydi bulmacanın ne kadar kolay çözüleceğini düşünün. O zaman problemin çözümüne ulaşmak için düşünmemize gerek bile kalmayacaktır. Bu yüzden farklı biçimde düşünebilen ve farklı çözüm yolları üretebilen kişiler bu problemi çözecektir.

Bu, problem çözmeye önemli bir ilkeyi göstermektedir: "Yapabileceğiniz olası tüm hareketleri öngöremezseniz sorunu çözemezsiniz.". Diğer yandan olası tüm işlemleri numaralandırarak, çalışan birleşimini bulana kadar farklı işlemleri deneyerek bir ok problemi çözebilirsiniz.

Kısıtlamaları listeleyelim. Buradaki anahtar kısıtlamalar şunlardır:

1. Çiftçi tekne içerisine tek seferde kendisi dışında yalnızca bir nesne daha alabilir.
2. Tilki ve kaz aynı kıyıda yalnız bırakılamaz.
3. Kaz ve mısır aynı kıyıda yalnız bırakılamaz.

Bu problem, kısıtlamaların önemini ortaya koyan iyi bir örnektir. Bu kısıtlamalardan herhangi birini kaldırırsak bulmaca kolay hale gelir. İlk kısıtlamayı kaldırırsak üç öğeyi tek bir turda alabiliriz.

İşlemleri listeleyelim.

1. İşlem: Kazı nehrin karşı tarafına taşıyın.
2. İşlem: Tekneyi bir kıyıdan diğerine götürün.
3. İşlem: Tilkiyi nehrin karşı tarafına taşıyın.
4. İşlem: Kazı nehrin karşı tarafına taşıyın.
5. İşlem: Mısır çuvalını nehrin karşı tarafına taşıyın.
6. İşlem: Tekneyi bir kıyıdan diğerine götürün.
7. İşlem: Kazı nehrin karşı tarafına taşıyın.

Bu Problemden Neler Öğrendik?

Sorunu daha biçimsel bir şekilde yeniden ifade etmek, bir problemi anlamak için mükemmel bir tekniktir. Birçok programcı, diğer programcıları bir sorunu tartışmak için arar; sadece diğer programcıların yanıtı olabileceğini düşünür fakat aynı zamanda problemi yüksek sesle ifade etmek genellikle yeni ve yararlı düşünceleri tetikler. Bir sorunun tekrar okunması, bu tartışmayı başka bir programcıya yaptırmak gibidir ancak her iki noktadan da destek alırsınız.

2.1.5.2. Sudoku

Sudoku oyunu gazete ve dergilerde yer almasıyla ayrıca web ve telefon tabanlı bir oyun olarak sunulmasıyla son derece popüler hale gelmiştir. Farklı varyasyonlar olmakla birlikte burada kısa olan geleneksel sürümü tartışılacaktır.

9x9 boyutlu bir tablo kısmen tek basamaklı (1-9) sayı ile doldurulur ve oyuncu belirli kısıtlamalara göre hareket ederken yalnızca boş kareleri doldurmalıdır. Her bir satır ve sütunda, her rakam tam olarak bir kez yazılmalıdır ve her doldurulmuş 3x3 alanda her bir rakam tam olarak bir kez yer almalıdır. O zaman, verilen sudoku yapısındaki boşlukları 1-9 arasındaki her bir sayıyı; bulunduğu satır, sütun ve kare içinde yalnızca bir kez kullanılacak biçimde nasıl doldururuz?

Sudoku bulmacanın zorluk derecesi doldurulması gereken karelerin sayısı ile belirlenir.

	9	1		6		7		
				8	2		3	9
5		3				2		
			9	1	3		6	2
		2	4		6	8		
1	4		8	2	5			
		9				5		7
6	7		1	5				
		5	4		6	9		

Soru şudur: Hangi kareleri ilk olarak doldurmaya çalışalım?

Bulmacanın kısıtlamalarını hatırlayın. Dokuz ana karenin her bir satır ve sütunda, her rakam tam olarak bir kez yazılmalıdır ve her doldurulmuş 3x3 alanda her bir rakam tam olarak bir kez yer almalıdır. Bu kurallar, çabalarımıza nereden başlamamız gerektiğini belirtir. Bulmacanın ortasındaki 3x3 alan, dokuz kareden sekizinde zaten sayıya sahip. Bu nedenle ortadaki kare, yalnızca 3x3 alanında başka bir karede temsil edilmeyen tek bir olası değeri içerebilir. İşte bulmacayı buradan çözmeye başlamalıyız. Bu alandaki eksik rakam 7'dir, bu nedenle orta kareye 7 rakamını yerleştireceğiz.

Bu değeri yerinde yazdığımızda, bu sütunun artık dokuz kareden yedisinde değerlere sahip olduğunu ve bunların her bir sütunda bulunmayan bir değeri olması gerektiği ve yalnızca iki kareyi boş bıraktığını unutmayın: İki boş karenin değeri 3 ve 9'dur. Bu sütundaki kısıtlama her iki numarayı da her iki yere yazmamıza izin verir ancak 3'ün üçüncü satırda, 9'un zaten yedinci satırda bulunduğuna dikkat edin. Bu nedenle satır kısıtlamaları, 9'un orta sütunun üçüncü satırına, 3'ün orta sütunun yedinci satırına yazılmasını gerektirir.

Tüm bulmacayı burada çözmeyeceğiz ancak bu ilk adımlar, ideal olarak mümkün olan en düşük sayıdaki kareler için aradığımız sadece bir tane önemli noktayı görmemizi sağlar.

Bu Problemden Neler Öğrendik?

Sudokunun temel dersi, problemin en kısıtlı bölümüne bakmamız gerektiğidir. Kısıtlamalar, çoğu zaman bir problemi zorlaştıran şeyken (Tilkiyi, kazı ve mısır çuvalını unutmayın.) aynı zamanda çözüm hakkındaki düşüncemizi basitleştirebilir çünkü seçenekleri ortadan kaldırır.

Yapay zekâyı bu derste özel olarak tartışmayacak olsak da, yapay zekâda "en katı değişken" olarak adlandırılan belirli türdeki problemleri çözmek için bir kural vardır. Kısıtlamaları karşılamak için farklı değişkenlere farklı değerler atamaya çalıştığınız bir problemde, en fazla kısıtlamaları olan değişkenle başlamak ya da olası değerlerin en düşük sayısına sahip değişkeni başka bir şekilde koymak gerekir.

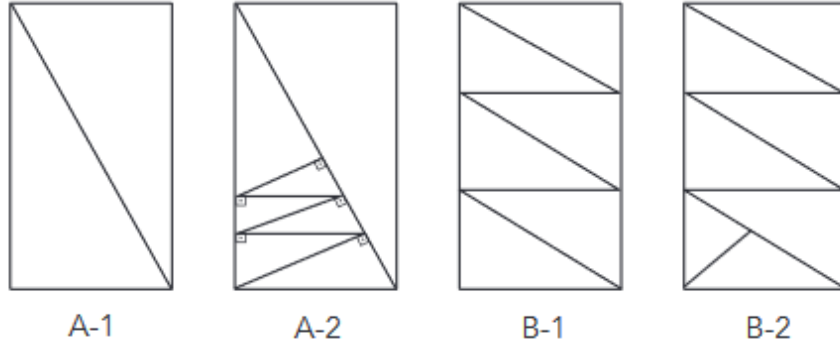
İşte bu tür düşünceye bir örnek; bir grup iş arkadaşınızın sizinle beraber öğle yemeğine gitmek istediğini ve herkesin beğeneceği bir restoran bulmanızı rica ettiğini varsayalım. Problem şu: Meslektaşların her biri grup kararında birtakım kısıtlamalar getiriyor; Ayşe vejetaryen, Can deniz ürünlerini sevmiyor, Serhan'ın çoklu gıda alerjisi var vb. Hedefiniz bir restoran bulmak için gereken süreyi en aza indirmek ise en sıkı kısıtlamalara sahip olan iş arkadaşınızla konuşarak başlayın. Örneğin Serhan'ın bir dizi geniş gıda alerjisi varsa Can'dan başlamak yerine, Serhan'ın yiyebileceği yemeklerin olduğu bir restoran listesi bularak başlamak daha mantıklı olacaktır. Deniz ürünlerinden hoşlanmaması daha kolay çözülebilir.

2.1.5.3. Dikdörtgenleri Parçalara Ayırma

Bir dikdörtgenden nasıl dik üçgenler oluşturabiliriz? Buna göre, bir dikdörtgeni bölerek, oluşturulabilecek dik üçgenleri şekil çizerek gösteriniz.

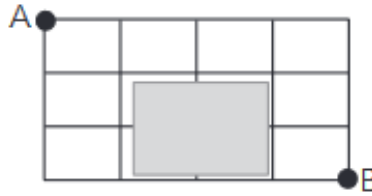
Bir dikdörtgenin kısa kenarı 1cm olduğunda uzun kenarının 1cm'den büyük olması gerekmektedir. Uzun kenar 2cm olduğunda A-1 şeklinde gösterilen dik üçgenler elde edilir. Oluşan dik üçgenlerden her birinin de A-2 şeklinde dik üçgenlere ayrılabiliriz.

İkinci çözüm yolu ise B-1 şeklindeki gibi dikdörtgen içerisinde birbirine paralel çizgiler oluşturmaktır. Oluşan küçük üçgenleri de B-2 şeklinde gösterilen biçimde dik üçgen olacak şekilde bölebiliriz.



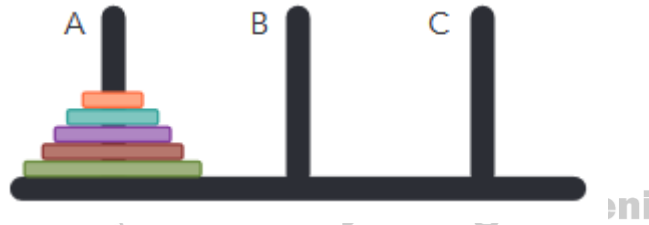
2.1.5.4. Engelli Yollar

Şekilde gösterilen A noktasından B noktasına gidebilmek için gri ile gösterilen alandan geçiş bulunmamaktadır. Buna göre A'dan B'ye gidebilmek için kaç farklı yol kullanılabilir?



Doğru cevap 17'dir. Sorunun çözümünde satır ve sütunların kesişim noktalarından yararlanılmıştır. Şekilde görüldüğü gibi 1. satır 1. sütundan başlanarak satır satır işlem yapılır ve her kesişim noktasında soldaki ve üstteki değerler toplanır. Kesişim noktasının üstünde ya da solunda değer yoksa aynı sayı tekrar yazılır. Örneğin 1. satır 1. sütundaki değer 1'dir. 1. satır 2. sütuna geçildiğinde sadece sol tarafta 1 değeri olduğu için 1 yazılır. 2. satır 2. sütunda ise kesişim noktasının solunda ve üstünde 1 değeri vardır. Bu nedenle bu kesişim noktasının değeri $1+1=2$ olur. En son 5. sütun 4. satıra gelindiğinde ise $9+8=17$ olur.

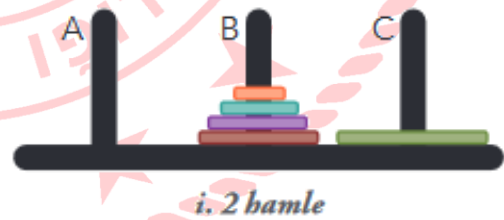
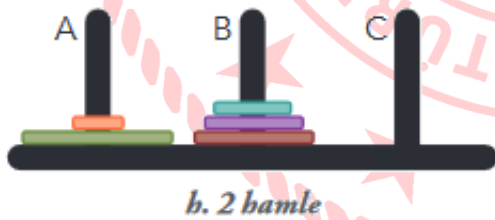
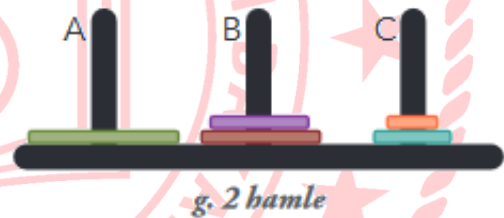
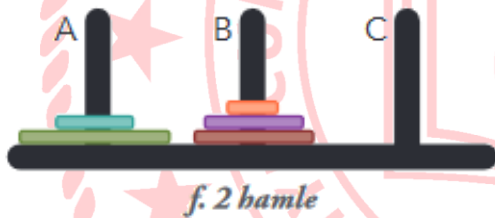
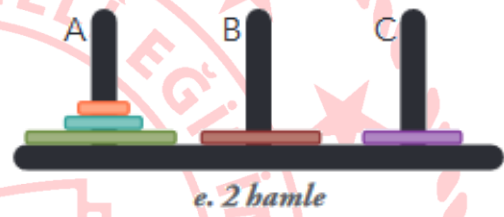
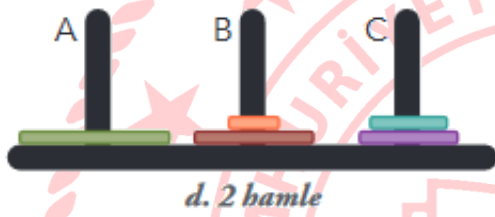
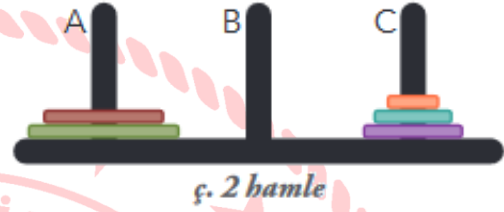
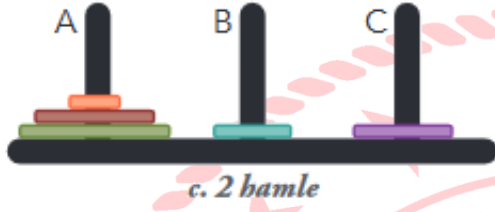
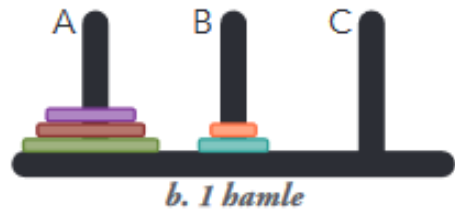
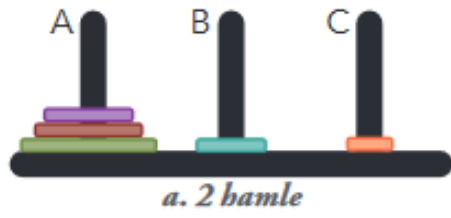
2.1.5.5. Hanoi Kulesi



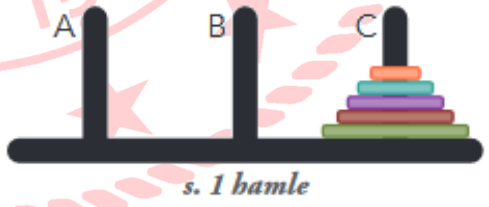
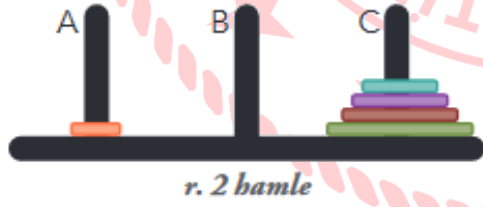
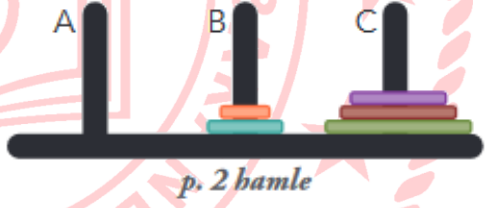
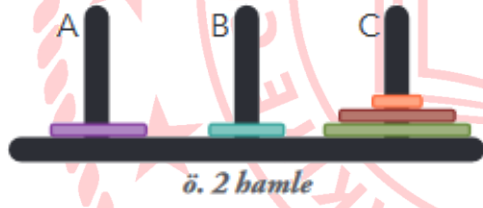
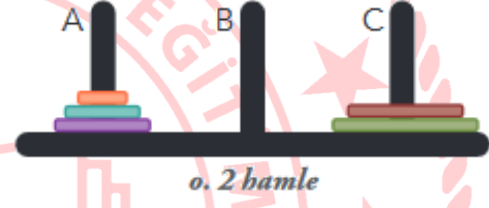
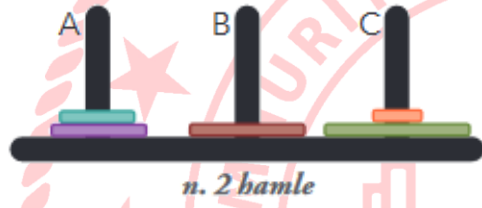
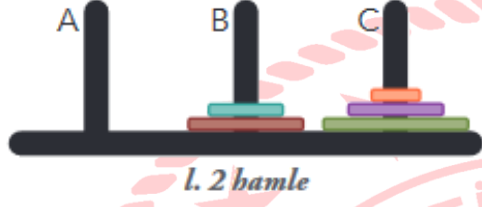
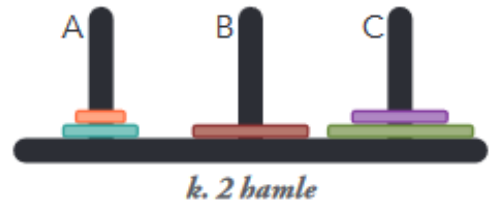
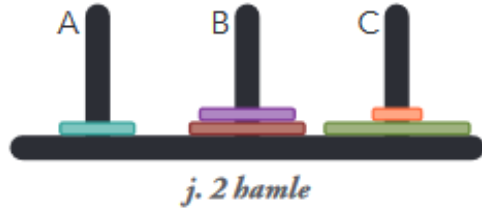
Yukarıdaki şekilde çeşitli renklerle gösterilen tahta parçaları, A çivisinden C çivisine aşağıdaki kurallara göre geçirilmek istenmektedir.

1. Küçük tahtaların üstüne büyük tahtalar yerleştirilemez.
2. Aynı anda sadece bir disk oynatılabilir.

Buna göre tahta parçalarını A'dan C'ye taşıyınız.



Ali GÖKAŞAR
Bilişim Teknolojileri Öğretmeni



Ali GÖKAŞAR
Bilişim Teknolojileri Öğretmeni