ALGORİTMA ÖDEV

# 1)Algoritma nedir? Algoritmanın gerekliliğini ve avantajlarını açıklayınız.

Algoritma nedir?

Bir sorunu, problemi çözmek veya belirli bir amaca ulaşmak için gerekli olan sıralı mantıksal adımların tümüne algortima denir. Ya da bir problemin çözümündeki adımların tümüne denir.

Bilgisayar düşünemez, bu yüzden algoritmanın her adımı anlaşılır, basit ve kesin bir biçimde ifade edilmiş olmalıdır. Yorum gerektirmemeli ve belirsiz ifadelere sahip olmamalıdır. Gereksiz tekrarlarda bulunmayan diğer algoritmalar içerisinde de kullanılabilir olmalıdır.

Her algoritmanın bir başlangıç noktası, belirli işlem adımı ve bir bitiş noktası içermelidir. Sonsuz döngüye girmemelidir. İşlem sonucu kesin olmalı, aynı veri için her yeni çalıştırmada aynı sonucu üretmelidir.

Algoritma neden gerekli ve avantajları nelerdir?

-Program yazmak kolaylaşır.

-Hatalı kodlama oranı asgari düzeye iner.

-Programın işlem akışı görülür. Böylece program kontrolü kolaylaşır.

-Program yazımı pratik hale geldiği için zaman kaybı olmaz.

-Sonradan programa ekstra özellik eklenmesi kolaylaşır.

-Program hızlı bir şekilde çalışır.

-Yazılan program bellekte fazla yer kaplamamalıdır.

Algoritmalar günlük hayatımızın her adımında vardır. Sayıların sıralanması, fotoğraf kamerasının yüz algılaması, Google ile kelime aratmaya kadar her şey bir algoritma mantığı üzerine kurulmuştur. Bir çay demlerken bile farkında olmadan kullanmakta olduğumuz algoritmaların önemini kavramak ve bu algoritmaların neler olduğunu bilmek çağımıza ayak uydurmak için gereklidir. Evde, işte ve hayatımızın diğer alanlarında karşılaştığımız problemleri etkin bir şekilde analiz ederek, çözüme en kısa yoldan en sade şekilde ulaşabilmek algoritmaları anlamak ve benimsemekten geçmektedir. Ayrıca özgün problemleri hızlıca çözüme ulaştırabilmek adına algoritma geliştirebilmek de algoritmaları benimsemeye bağlıdır.

# 2)Program yazılmadan önce algoritmamı hazırlanmalı yoksa akış diyagramı mı çizilmeli?

Akış diyagramı: Algoritmanın görsel olarak simge ya da sembollerle ifade edilmiş şekline ‘akış şemaları’ veya FLOWCHART adı verilir.

Akış şemalarının algoritmadan farkı, adımları simgeler şeklinde kutular içine yazılmış olması ve adımlar arasındaki ilişkilerin ve yönün oklar ile gösterilmesidir.

Akış diyagramı çizilmelidir. Akış diyagramında kullanılan şekillerin uluslararası geçerliliği olduğundan, herkes tarafından rahatça anlaşılabilir.

3)Algoritma hazırlanırken dikkat edilmesi gereken hususları açıklayınız.

-İlgili adım, herkesin anlayabileceği gibi net bir şekilde ifade edilmelidir.

-Probleme uygun olarak algoritma adımları arasındaki işlem sırasına dikkat edilmelidir.

-Tüm olası durumları içerecek kadar genel olmalıdır.

- Başlangıç-bitiş adımlarını içermeli, sonlu sayıda işlem adımına sahip olmalı ve belirli bir sürede bitmelidir.

-Problem çözümü herkes tarafından anlaşılabildiği için takım çalışmasını da kolaylaştırır.

-Projeyi tasarlayan haricinde bir başka kişi tarafından da kodu yazılabilir.

-Kodlama öncesi kontrollerin yapılarak mantıksal hataların tespit edilmesini kolaylaştırır.

# 4)Değişken nedir? Programlamada neden değişkenlere ihtiyaç duyulmaktadır?

Değişken: Hafızada bir yer ayırmak, gerekli durumlarda ayrılan yere değer atamak, değeri değiştirmek ve okumak için kullanılan programlama bileşeni değişken olarak adlandırılır.

-Bir başka ifadeyle değişken, bir değeri tutan depolama konumudur.

-Değişken, üç temel boyutu ile düşünülmelidir;

\*Değişkenler, RAM bellek ünitesinde tutulduğundan geçicidir ve üzerindeki değerler sürekli değişebildiğinden bu isimle anılmaktadır.

\*Program yazılırken değişken tanımlandığında hafızada, RAM ünitesinde değişken için bir alan ayrılır.

\*Ayrılan alanın büyüklüğü değişkenin tipine ve sakladığı değere göre değişiklilik göstermektedir.

\*Değişkenler tanımlanırken bazı kurallara uyma zorunluluğu varıdır.

\*C# dilinin kullandığı bağzı tanımlayıcılar, anahtar kelime olarak adlandırılır. Bu nedenle bu anahtar kelimeler değişken ismi olarak kullanılamazlar.

\*Değişken isimleri harf ile başlamalıdır. Rakam ile başlayan değişken isimleri programın hata vermesine neden olur.

Programlamada değişkenlere ihtiyaç duyulma sebebi: C# programlama dilinde değişkenler, bir değeri veya nesneyi saklamak ve programda bu değeri kullanmak için kullanılır. Programcılar değişkenleri kullanarak programda işlem yapmak için gerekli olan verileri saklayabilir ve bu verilere programın herhangi bir yerinde kolayca erişebilirler.

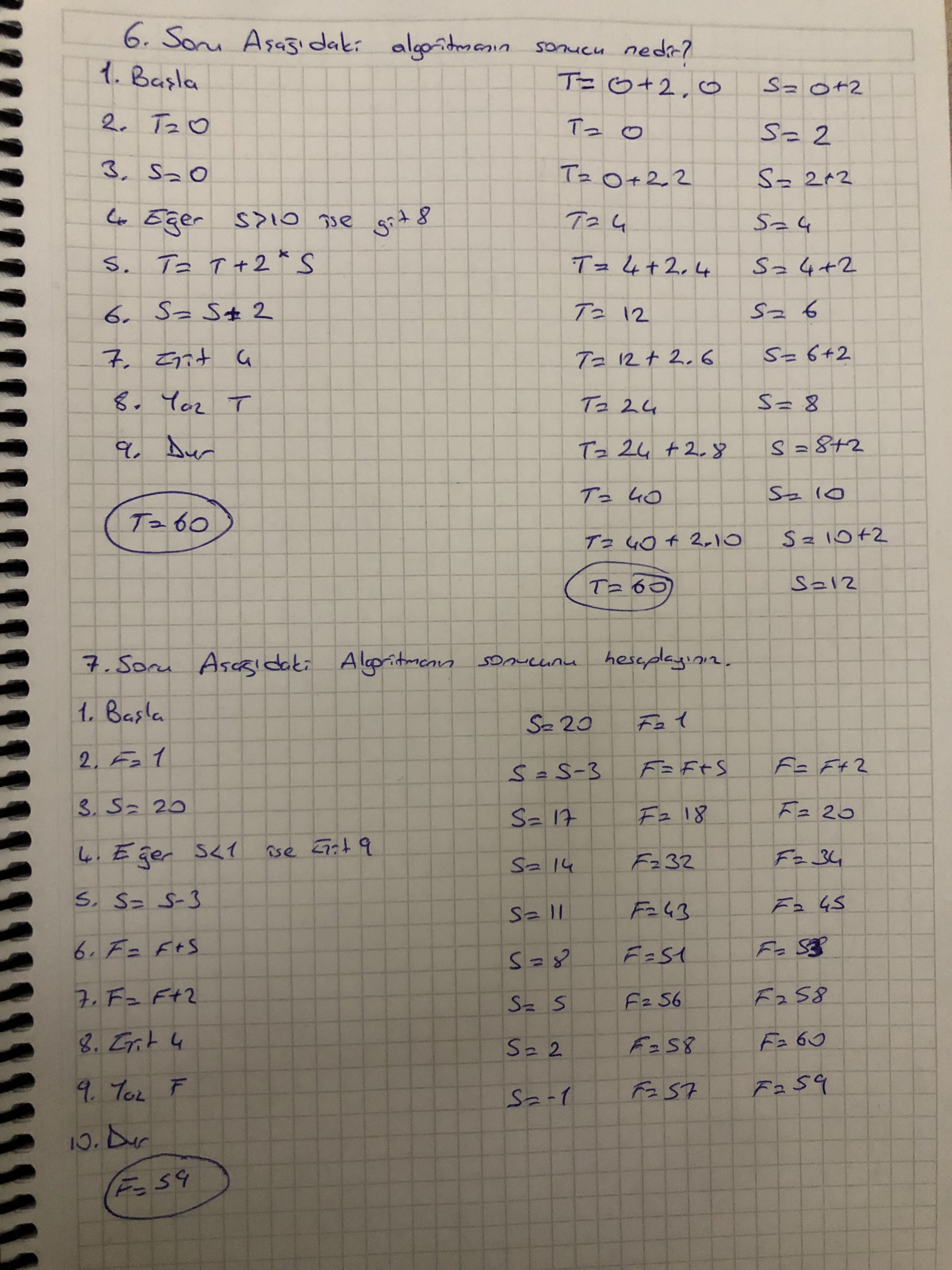
\*C# programcılarına programlarında dinamiklik kazandırmak için önemlidir. Örneğin, kullanıcı tarafından girilen verileri program içinde saklamak içinde saklamak için değişkenler kullanılabilir. Bu sayede program, kullanıcının girdiği verileri saklayabilir ve programın diğer bölümlerinde bu verileri kullanarak işlem yapabilir.

5)Sayaçlar nerelerde ve niçin kullanılmaktadır?

**Sayaç:** Programlarda; bazı işlemlerin belirli sayıda yapılması veya işlenen/üretilen değerlerin sayılması gerekebilir. Örneğin klavyeden girilen bir cümlede kaç tane sesli harf olduğunu bulan programda, cümlenin her harfi sırasıyla çağırılır ve sesli harfler kümesine ait olup olmadığı araştırılır.

Eğer sıradaki çağırılan harf bu kümeye ait ise, bunları sayacak olan değişkenin değeri bir arttırılır. Say = Say + 1şeklindeki işlemde sağdaki ifadede değişkenin eski (mevcut/önceki) değerine ‘1’ eklenmekte; bulunan sonuç yine kendisine, yeni değer olarak aktarılmaktadır.

Bu tür sayma işlemlerine, algoritmada ‘’sayaç’’ veya ‘’sayıcı’’ adı verilir. Yani ‘’sayaç’’; işlem akışı kendisine her geldiğinde, belirtilen adım değeri kadar artan/azalan değişkendir. Sayaç değişkeni = sayaç değişkeni ± adım Örneğin X=X+3 üçer üçer artan, S=S-5 beşer beşer azalan bir sayıcıdır.



8)Girilen üç sayıdan en büyüğünü bulan programın algoritmasını hazırlayınız.

1. BAŞLA

2. 1.sayıyı gir. (a)

3. 2. sayıyı gir. (b)

4. 3.sayıyı gir. (c)

5. Eğer (a>b) ve (a>c) ise

Yaz; 'büyük olan sayı'' a

Eğer bitti.

6. Eğer (b>a) ve (b>c) ise

Yaz; ''büyük olan sayı'' b

Eğer bitti.

7.Eğer (c>a) ve (c>b) ise

Yaz; ''büyük olan sayı'' c

Eğer bitti.

8. DUR.

9)Girilen üç sayıyı küçükten büyüğe doğru sıralayan programın algoritmasını hazırlayınız.

1. BAŞLA

2. 1.sayıyı gir. (x)

3. 2.sayıyı gir. (y)

4. 3.sayıyı gir. (z)

5. x>y ve y>z ise

x>y>z mesajını yazdır git 11

6. y>x ve x>z ise

y>x>z mesajını yazdır git 11

7. z>x ve y>x ise

z>y>x mesajını yazdır git 11

8. x>z ve z>y ise

x>z>y mesajını yazdır git 11

9. y>x ve z>x ise

y>z>x mesajını yazdır git 11

10. z>x ve x>y ise

z>x>y mesajını yazdır git 11

11. DUR.

10) 1-99 arasındaki tek ve çift sayıların toplamları ile çarpımları ayrı ayrı hesaplayan programın algoritmasını hazırlayınız.

1. BAŞLA.  
 2. tektop, tekçarpım, çifttop, çiftçarpım, döngü=1;  
 3. Döngüyü başlat (döngü<100)  
 4. EĞER döngü/2==0 ise cifttop=çifttop+dongu ve çiftçarpım=çiftçarpım\*dongu işlemini yap.  
 5. EĞER döngü/2==1 ise tektop=tektop+döngü ve tekçarpım = tekçarpım\*dongu işlemini yap.  
 6. döngü++.  
 7. döngü<100 ise Adım 4'e git.  
 8. DUR.