Computational Thinking Nedir (komputeşınıl tinking) (sayısal, hesaplamalı düşünme)

Bizim karmaşık bir problemi alıp bu problemi küçük parçalara ayırıp, anlayıp bunun üzerine çözüm üretmemiz demektir.

Bilgisayar insandan farklı çalışıyor. Biz bilgisayara bir şey çözdüreceksek bilgisayarın düşünme tarzında yapmamız gerekiyordu. Algoritmayı anlatırken bahsetmiştik…

Neydi bu;

* Detaylı tanımlamalar
* Düzgün sıralanmış komut setleri
* Problemin uygun bir şekilde fonksiyonlara göre tanımlanmasıydı. (matamatik fonk olabilir)

CT in ( Bunun ) alt parçaları var

* Dikompuzeşin------- Bir problemi daha kolay yönetilebilen ufak parçalara ayırma
* Pattern leri tanımlayabilme bunları fark edebilme önemli olacak ( patern tanımlaması-----örüntü gibi 1 3 5 7 gibi)
* Abstrakşın----Özetleme, soyutlama anlamlarına gelir, yani gereksiz şeylerden kurtulup, sadece çözmeye çalıştığımız problemi önemli detaylarına konsantre olmak, odaklanmadır.
* Algoritma ise herhangi bir işi başarmak için detaylı bir şekilde tanımlanmış adımlar dizisidir

Bunlar 4 önemli başlık

Preclas okuduk mu

Tahmin etmeye ve tartışmaya çalışalım

CT bu ders biraz etkileşimli bir ders ara ara girdilerinizi isteyeceğim

CT , bizim şu ana kadar hayatımızda gördüğümüz başımızdan geçen nelere benziyor, bunu daha önce duydunuz mu ?

( sınava çalışmadan önce nelere çalışacağız ------4 ana başlıkta topladığımız adımları yapıyoruz)

(ev toplamak taşınırken ---1- odaları ayrı ayrı toplama. 2- kırılacakları ayrı koymak, yorganı ayrı koymak gibi. 3- eşya taşırken o gün futbol maçına gitmez, misafir çağırmaz. 4. Önceliklerimizi sıralarız eve gidince ilk neler yapılacak. Problemimizin detaylarına göre sıralamaktır.)

(araba kullanmak

1. Önce arabaya binmeden yapılacakları kontrol ederim, daha sonra arabaya bindikten sonra ne yapacağım onlara bakarım, problemi compozlara bölerim
2. Manuel araba mı otomatik vites mi kullanacağım
3. Telefonla konuşmak olabilir, ilgisiz ve alakasız konular olabilir
4. Bunları yaparken neyi neyden önce veya sonra yapacağım onu belirlerim, anahtarı çevir, debriyaja bas, frene bas, gaza bas şeklinde ilerlerim.

Genelde tekrar tekrar yapılacak işlerde bilgisayar teknolojisini kullanırız.

Compiteşınıl tiiking ; biraz daha incelediğimizde

* Locig, mantıksal düşünme
* Kritik düşünme
* Problem çözümünün ön plana çıktığını görüyoruz

Bunları biraz daha açacak olursak

1. Bilgisayarda logic dediğimizde aklımıza ilk gelen şunlardır. 1/0, true/false, </>, +/- mi

Yani düz vites mi otomatik mi, lastikler sağlam mı değil mi, paketlediğimiz eşyalar kutuya sığacak mı sığmayacak mı gibi

Veya lastik and yakıt var mı

Yağmurlu mu hava şemsiye or yağmurluk

Logic dediğimiz, yani problemi incelerken , çözerken bu esnada kararlar alırken bunları daima mantıksal bir zincir içinde düşünmeliyiz.

Hepimiz mantık dersleri aldık değil mi ve veya nın ne olduğunu biliyoruz.

1. Kritikıl tiking ise, problemin kendisiyle uğraşmak, problemin kritik yerlerini tespit edebilmek

Örneğin siz bir bankaya para yatırma problemi çözerken buradaki en önemli unsur ne (yatırılacak para yokken bankaya gidiyorum, kritik unsuru atladığım için bu problemi çözemem) diğerleri olsa bile bankaya gittim, giriş işlemleri logic im tam olsa bile çözemem.

Veya bankadan önce markete gittim , alış-veriş yaptım 10 saat ne oldu banka kapandı.

1. Aynı zamanda benim problemin çözümüne uygun adımları yapmalıyım

Parayı aldım, bankaya gittim ama yanlış bankaya gittim. Yani buradaki problem çözümünde benim amacıma uygun hareket etmeliyim.

Problem çok zor ise ne yapmam gerekir ?

Parçalara bölmek, gereksiz şeylerden kurtulabilir, iş bölümü yapmak, yardım almak gibi.

Başlamadan Önce Düşünülmesi Gereken Bazı Sorular

1. Bir çok probleme aynı çözümler varsa bu bize ne söyler (10 problem var 2 çözüm var bu problemin 8 çözümüne de bu iki çözümle yapabiliyorsam ne yaparım)

Evde bir sürü tamir problemim var ben yıldız tornavidayı kullanırsam bir çok tamir işini bu şekilde halledebilirim.

Veya bir oyun şirketinde çalışıyorsunuz----kullanıcının davranış analizini yapmanızı istiyorlar, ben ne sunarsam daha çok bundan nasıl para kazanabilirim diyor biz burada bulacağımız çözüm hep aynı para kazanmak, ne yapabiliriz bunun. Kendime uygun çözümler belirleyip fonk yazacağım ve kütüphaneler yazacağım ve ne zaman bu problemle karşılaştığımda bunu kullanırım.

1. Eğer bir problemin varsa ve bu problem senin çözümünü yaptığın, bildiğin bir başka problemden çok azıcık farklıysa ne yaparsınız.

İki problem birbirine çok yakın ne yaparım.

İki problem arasındaki pattern i ben tespit ettiysem ilk problem 100 e kadar toplama problemi, ikici 50 tane , bu iki problem arasında pattern farklılıkları var. İlk problemdeki çözümümü alırım ikince problemdeki çözümüme adapte ederim.

Bundan sonra yazdığımız kodları, öğrendiğimiz çözüm toollarını uygun bir şekilde tasnif ederseniz ve açıklamalarını yazarsanız, yeni problemlere karşı etkin çözümler geliştirebilirsiniz.

Daha önceki derslerden ayrı bir şey değil üzerine ekliyoruz şimdi.

1. Dikompozişın---- parçalara ayırmak diyor
2. Patern rokkogeşin--- paternleri ve modelleri yorumlamak
3. Abstrakşın ----------için ne diyorduk ilgimiz olmayan şeylerden kurtulma diyoruz. İlgimiz olmayan şeylerle uğraşmıyoruz. Bunun için verilerimizi tasnif etme, organize etme kullanılabilir. Aynı şekilde bilgilerimizi mantıksal bir şekilde de özetleyebiliriz, tasnifleyebiliriz, önemsiz şeylerden kurtulabiliriz.

Tasnif herhangi bir nesne, durum veya olay örgüsünü, bölümlendirme, sıralama veya kategorize etme özelliklerini kullanır. Bir karmaşanın önüne geçilmesi maksadıyla tasnifleme işi yapılır.

1. Algoritmalar da, bir algoritmayı tasarlama ve onu implement etme, bulduğunuz çözümü koda dökmedir.

Programlama Nedir ?

Çözümü planlama, işlem adımları belirleme ve uygulama, bir problemin çözümü için mantıklı ve sıralı adımlar

Compiteşınıl tiking nedir

Bu bilgiler ışığında geldiğimiz çözüm şu

* CT, bir problem sahibi olma ve bu probleme çözüm üretme arasındaki olması gereken adımdır.

Evet benim bir problemim varsa ve buna bir çözüm üreteceksem CT bir gerekliliktir. Bu da nedir ; programlama= algoritmalar + coding demiştik.

* CT in amacı ise bir bilgisayara programlanabilecek bir çözümün nasıl oluşturulacağını öğretmeyi amaçlar.

Mimarlar mı Duvarcılar mı?

Programlama tuğla örmeye çok benzer: bir şey inşa edebilmek için onun hakkında biraz bilgi sahibi olmanız gerekir ve usta zanaatkarlar işaretleme ve balıksırtı düzenlerindeki varyasyonların göreli avantajları hakkında uzun ve ayrıntılı konuşmalar yapabilir, ancak temelde bir duvar bir duvardır. .

Tuğla örmek o kadar da ilginç değil: mimari ilginç. Nasıl mimarlık, insanların gereksinimlerini anlamak ve özellikle şekillendirilmiş bir tuğla yığınının onlara nasıl hitap edebileceğini görmekle ilgiliyse, CT de bir sorunu anlamak ve özellikle şekillendirilmiş bir program ifadeleri yığınının buna nasıl hitap edebileceğini görmekle ilgilidir.

CT in 4 dayanağı, sütunu;

Dikompozeşın

Temel olarak problemi küçük parçalara bölmek anlamına geldiğini söylemiştik.

Zor bir sorunumuz var en başta, biz bu zor sorunu 3 parçaya ayırıyoruz. 2 kolay 1 zor parçaya ayırıyoruz. Daha sonra bu zor olanı da 2 kolay parçalara ayırıyorum.

Örnek verelim daha öncekilerden benzetebiliriz. (ev taşıma da mesela)(ders çalışma gibi kolaylar ve zorlar) (evi inşa etmek gibi)

\*\*

Verileri, süreçleri veya sorunları daha küçük, yönetilebilir parçalara ayırma

Ayrıştırma, proje planlamasından daha fazlası için kullanılır. Belirlemek için çok büyük veya hantal olan karmaşık kavramları anlamada kullanışlıdır. Bir kavram ayrıştırıldığında, onu oluşturan parçaların incelenmesi, konuya yaklaşmak için yeni yollar sunabilir. Süreçleri optimize edebilir, tasarımları kolaylaştırabilir ve hatta tamamen yeni fikirler üretebilirsiniz.

Bir problem ayrıştırılmazsa, çözülmesi çok daha zordur. Birçok farklı aşamayla aynı anda uğraşmak, bir problemi daha küçük problemlere bölüp her birini teker teker çözmekten çok daha zordur. Problemi daha küçük parçalara bölmek, her küçük problemin daha detaylı incelenebileceği anlamına gelir.

Dikompozeşin da ağaç yapılarını kullanıyoruz, gözümüzde canlandırmak için.

Burada aktiviteler var

diyagram içeren bir resim

Açıklama otomatik olarak oluşturuldu

* Ağaç yapıları, problemlere alt problemlere ayırmamıza yardımcı olur.
* Kollarla birbirine bağlanan bloklardan oluşur.
* Üst geçidine bağlı olan her bloğu, o geçidin bir alt problemini temsil eder.

Problemi bu şekilde analiz etme alışkanlığı kazanırsanız, size çözümlerinizin çok daha etkili ve kolay olmasını sağlayacaktır.

Örneğin çikolatalı kek yapmak ise problemimiz, bakalım bunu nasıl dikompoz yapabiliyoruz. (problemin çözümünü reflex haline getirenlerde olabilir)

* İçine konulacak malzemelerin hazırlanması

Önce eksikleri satın almaya gitmiş, zaten elinde olanlara bakmış. (önce elimde neler var eksikler ne ise alayım şeklinde olabilir.)

* Malzeme ve pişirme ortamlarının ayarlanması olarak ikiye bölmüş. ( çevre ve malzeme)

Ortam olarak, gereken ortam olarak fırını ısıt, mutfak tezgahını boşalt veya temizle, malzeme olarak da çırpıcı ve karıştırılacak kab ı hazırla şeklinde oluyor.

Burada çikolatalı kek yapımını dikompoz etmiş.

Biz burada neye dikkat etmeliyiz, (algoritma) sıralamaya dikkat etmezsek programımızda ne olacak, (bag ) programda hataya sebep olacak.

\*\*Arabayı parketmek için;

\*\*Seyahat edeceksiniz, planlarınızı neye göre yaparsınız ?

Burada yeni bir çalınç daha var, Zihin haritası şeklinde karalama yaparak, merkez fikir ortada kenarlaraına aklınıza gerekenleri karalayarak bir problem çözüyoruz.

\*\*İşe gideceğiz, bunu dikompoz edelim

--Evden çıkmadan önce

--kıyafet

--çanta

--hangi araç ve yol güzergahı

metin, elektronik donanım, yazıcı, ekran görüntüsü içeren bir resim

Açıklama otomatik olarak oluşturuldu

\*\*Bir IT uzmanı iseniz;

Burada sizlerle bir aktivite yapacağız, herkesin kendi çözümünü yazmanızı isteyeceğim.

Siz bir tur rehberisiniz ve burada da şehrin bir haritası var. Sizin rehberliğinizi yapacağınız turistler şehrin görülecek yerlerini otelden başlayarak gezmek istiyorlar.

Ama her yeri gezmeniz, hiçbir yere iki defa gitmemeniz ve en sonunda da bu seyahati otelde bitirmeniz gerekiyor.

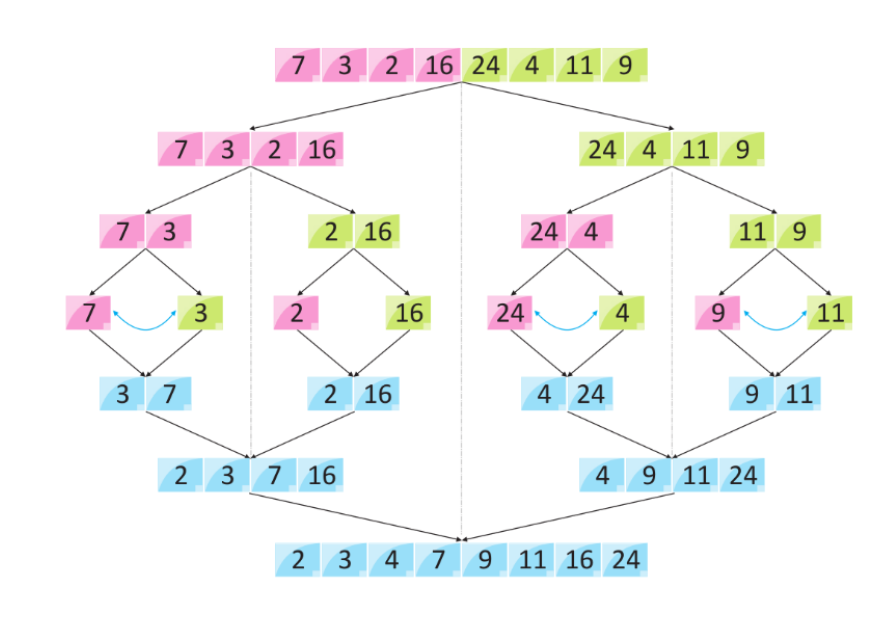
Otel 2 numara , 2 numara ile başlıyoruz. Sizden yazmanızı istediğim cevap şu 2 den başlayıp numaralar halinde belirtmeniz.

2-5-6-8-7-4-1-10-11-9-3-2

Problemimin iki veya daha farklı çözümü olabilir. Eğer aralarında bir fark yoksa, kullanıcı ihtiyaçlarını gideriyorsa bu da çözüm olabilir.

\*\*sayılar arasında da sıralama işlemleri yaparken nasıl dikompoz ediyoruz

7 3 2 16 24 4 11 9



Pattern rikognişın

Hesaplamalı düşünmede örüntü tanımanın ne anlama geldiğini tartışalım ve tahmin etmeye çalışalım!

Örüntü tanıma iyidir çünkü aynı sorunla her karşılaştığımızda yeni bir çözüm bulmak zorunda değiliz. Bunun yerine, önceden tanıdığımız kalıpları uyguluyoruz.

Computeşinıl tiking kullanarak problem çözme ihtiyaçlarınızda kullanacağınız en önemli malzeme bu diyebiiriz.

Sebebi şu;

Örüntü tanıma, farklı şeylerdeki (çözümlerdeki) örüntüleri tanımak ve bu örüntüleri diğer benzer problemlerin çözümlerinde uygulayabilmektir.

Genel olarak sayılarla, verilerle, analizlerle, sistemlerle uğraşacaksınız. Eğer sistemler arasındaki benzerliği çözerseniz (örneğin Linux, mac, Windows var ,bunlar arasındaki benzerlikleri çözerseniz sizin hangisini kullanacağınızı , problem çözmede hangisini kullanacağınıza daha iyi karar verirsiniz.

Patern tanımlama , paternleri fark etme bize şu açıdan işe yarayacak, her seferinde aynı sorunla karşılaştığımızda yeni bir çözüm bulmak zorunda kalmayacağız. Bunun yerine, tanımladığımız kalıpları uygulayacağız. Bu sayede vakit kazanacağım, kaynak kazanacağım, eski tecrübelerimi kullanacağım. Bu sayede üretimi hızlandıracağım.

Neyi fark etmemiz gerekiyor, denizaltı da olsa, araba da olsa, uçak da olsa etrafındaki akışın benzer ve önemli olduğunu farketmeli..

1 den 200 e kadar olan sayıları toplayalım

Bakalım Burada sorunumuz bu ? kim yorum yapmak ister

Büyük sorun?

Denedin mi?

Kafanda yapacak çok şey var mı?

1. 1+2+3+4+5+6+…..+200
2. Birbirine benzeyenleri tespit ederim veya çiftleri tekleri toplarım
3. Matamatik formülünü bulurum---- Google den bakabilirim

Burada ne yaparım hemen 3 tane çözümü yazabilirim.

Buradaki paternimiz nedir ?

N(n+1)/2 şeklinde toplama işlemine patern, örüntü kullanılarak formül yazılmış

Bu neden önemli ?

Karşılaştığınız problemlerde bu şekilde sizin önünüzdeki veriyi analiz edip bu benzerlikleri bulmanız gerekecek, o zaman sizin bu probleme çözümünüz matematiksel bir temele dayanacağı için hem programınız basitleşecek, hem çözümünüz daha doğru olacak, hem de işleminiz matematiksel bir zemine dayandığı için hata olmayacak.

Örnekleri inceleyelim

Add 1, 3, 5, 7, 9 diye ekleme yaparak ilerliyor

Buradaki patern nedir

Önceki digitlerin toplamını çıkart şeklinde bir paternimiz var. Görmesi daha zor problemlerimiz de olabilir.

Buradaki patern nedir

1 2 4 8 16 32 64 128 x2 şeklinde ilerliyor

Buradaki patern nedir

Farklı düşünün, kalıpların dışına çıkın, kutunun dışına çıkın problem basit ama, toplama çıkarma yapmayın

84935276 84935276 84935276

Daima serbest düşüneceğiz, bazen öyle problemlerle karşılaşacaksınız ki günlerinizi alacak, yemek yerken, yürürken aklınıza gelecek ne yapmalı diye işte o zaman ara vereceksiniz, daha sonra düşünmeye devam edeceksiniz. Birgün uğraşırken bir anda aklınıza gelecek ve oradan çözüme yürüyeceksiniz.

Yıldızların oluşumu her satırda artıyor bu nasıl oluyor ?

2n-1 formülüyle bulunur

Aklımda sayı tutma oyunu nasıl oluyor ?

Kuralları bulmak için her iki kullanıcı deneyiminde de tekrar eden bir kalıp bulduk:

“0 ile 10 arasında bir sayı tahmin ediyorum”, “tahmin etmeye çalıştı da…” dedi, “dedim…” gibi ifadeler tekrarlanır. Yani oyunu oynayabilmek için bu cümleleri kullanabiliriz. Bunun dışında tahmin ettiğimiz sayılar ve tahmin edilen sayının oyuncunun düşündüğü sayının altında mı üstünde mi olduğu ile ilgili ifadeler gibi bölümleri değiştiriyoruz.

Abstrakşın

Gereksiz olan ve problemin çözümüne katkı sağlamayacak olan bilgilerden kurtulmak.

Bir program yapıyorsunuz, verilere baktığınızda Sizin hiçbir şekilde probleme katkısı olmayacağını düşündüğünüz verileri en baştan ayıklamanız gerekir.

Neden , ya tamam kalsın onların bir zararı olmaz, zaten problemin çözümünü etkilemeyecek.

Problemi anlamanız zorlaşır, algoritmanız karmaşıklaşır, programınız karmaşıklaşır, gereksiz hafıza kullanırsınız, ayağınıza bağ başınıza dert edersiniz.

Gereksizse çözüme katmayacağız, gerekliyse koyacağız.

Soyutlama, çözüme hiçbir katkısı olmayacak gereksiz bilgilerden kurtulmaktır.

Soyutlama, bilgisayar bilimi ve hesaplamalı düşünmenin temel kavramıdır. Gerçek bir dünya probleminin bilgisayara ifade edilebilmesi için problemin soyut olması gerekir.

Bir alış-veriş de bir şey ararken filtreleyebiliriz.

Haritaya bize lazım olan şekliyle bakmalıyız yoksa gideceğimiz yeri anlamamız zorlaşır

LET’S

Bakalım bizim neleri abstrakt edip neleri etmeyeceğimizi görelim

Buradaki önemli olan gereksiz olanları ignore ( göz ardı etmek) ve faydalı olanları dahil etmek.

O zaman aşağıdaki örneklerden gereksiz olanları çıkartalım, sadece gerekli olanları bırakalım.

Ben araba dan neleri abstract etmeliyim neler kalmalı------

Kalem

Telefon

Ev----ev benim için sığınaksa geri kalan kapı, pencere benim için abstrakt tır

Mümkün olduğunca soyutlamayı deneyin. (Örneğin soyutlama olarak baktığınızda ev insanlar için sığınaktan başka bir şey değildir)

Burada aklınızda tutacağınız, size yardımcı olacak düşünce tarzı ise; bu objenin çözdüğü probleme olan katkısını düşünün

Muğlak bir ifadeyle program, algoritma yapamayız. Net şeyler olmalı.

Ev den mesela, koltuk, ayna, masa çıkmalı, neler kalmalı kapı, pencere, ısıtma sistemi olmalı.

Araba-----önemli----- motor, lastikler, cant, koltuk

Önemli değil-----renk, şekli, bagajı, radio

Kalem------ucu ve düzgün tutulabiliyor mu---------malzemesi, tipi, markası öneöli değil

Telefon-----hoparlör, mikrofon, sim kartı olacak

Önemli değil-----renk, markası, kamera

\*

Burada çok farklı cevaplar verebiliriz bunu gördük, bizi bağlayıcı olan nedir , problemimizin tanımı kapsamında kullanımlarımızın ihtiyaçları kapsamında bir abstrakçın a gitmek olacak.

Son olarak, soyutladığınız nesneleri, göz ardı edilmesi gerektiğini düşündüğünüz kısımları göz ardı ederek ve dahil edilmesi gerektiğini düşündüğünüz kısımlara dahil ederek tanımlamaya çalışın.

Video izle

<https://www.google.com/search?q=best+of+digital+literacy+computational+thinking+for+children&sxsrf=APwXEdeucPREWszEiNyfi-hMYrEeWcV6mw:1680629158852&source=lnms&tbm=vid&sa=X&ved=2ahUKEwjqxYzA35D-AhXXQvEDHQ-FDCsQ_AUoAXoECAEQAw&biw=1536&bih=722&dpr=1.25#fpstate=ive&vld=cid:3480965e,vid:mUXo-S7gzds>

KAHOT

1. Komputişinıl tiking nedir

* Bilgisayar gibi düşünmektir.
* Problem çözme tekniğidir\*\*\*\*\*\*\*\*\*\*\*\*
* Bilgisayarı programlamaktır.
* Bilgisayarla bir konuşma yöntemidir.

1. Hangisi yanlıştır

* CT bir problem çözme yöntemidir
* CT bilgisayar programları için önemlidir
* CT bize programlama dillerini nasıl öğreneceğimizi öğretir\*\*\*\*\*\*\*
* CT her hangi bir problemin çözümünde faydalıdır

1. Hangi meslek CT kullanabilir

* Bilgisayar müh
* Emniyet güçleri
* Tesisatçılar
* Hepsi

1. Hangisi CT in 4 konseptinden biri değildir
2. Dicompozişın a örnek hangisidir

* Bir yemek yapımının detayını adım adım çıkarmak-------algoritma
* Bir şehir haritası yapmak, sadece yolları içeren, binaları içermeyen ------abs
* Araba yıkamayı parçalara bölmek, iç ve dış olarak \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*
* Farklı yemekleri pişirirken ortak paternleri tespit etmek

1. Patern rekıgneşin en çok neye yardım eder

* Daha önceden hiç kullanılmamış bir çözümü bulmaya
* Farklı problemlere benzer çözümler bulmaya\*\*\*\*\*\*\*\*\*\*\*\*
* Problemin detaylarını anlamak
* Bir probleme en iyi çözüm olabilecek paterni göstermek

1. Abstrakşın a örnek hangisidir

Basit olmalı

1. İyi bir algoritma nasıl olmalı

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Algoritma adımına bakacak olursak bunu daha önce görmüştük.

Algoritma sözü, Özbekistan'ın Harezm, şimdiki Türkmenistan'ın Hive doğumundan doğan Ebu Abdullah Muhammed İbn Musa el Harezmi'den gelir . Bu alim 9. yüzyılda cebir alanındaki algoritmik çalışmalar kitaba dökerek matematiğe çok büyük bir katkı sağladı.

* Algoritma, çok net tanımlanmış ve bir akışı tanımlayan, bir işlem sırasını tanımlayan sıraydı.
* Algoritmalar ne zaman kullanışlıdır, birisine veya bir bilgisayara adımları nasıl yapacağımızı anlatmak için kullanışlı şekillerdir, grafiklerdir.
* Algoritmalar programlama için önemlidir çünkü doğru bir algoritma bilgisayar temelli çözümün odak noktası, başlangıcı, kaynağı , her şeyidir.

Algoritma, bir süreci tanımlamak için açıkça tanımlanmış adımlar dizisidir.

Algoritmalar, bir başkasına veya bir bilgisayara adımları nasıl gerçekleştireceğimizi açıklamak istediğimizde faydalıdır.

Algoritmalar programlama için önemlidir çünkü doğru bir algoritma herhangi bir bilgisayar tabanlı çözümün nihai temelidir.

Örnek

Çoban, kurt, kuzu, ot ve kayık karşıya geçecek. Her kayığa sadece iki kişi binecek(tabi kısıtlamalarımız var)

1. Adım; kuzu gitsin
2. Adım; ot gitsin
3. Kuzu gelsin
4. Kurt gitsin
5. kuzu tekrar gitsin

Mutlaka bir problemi çözerken algoritma çizin.

Kahve demleme algoritmanızı çizmenizi isteyeceğim

1. başla
2. kaç kişilik
3. şekerli / şekersiz mi
4. sütlü / sütsüz mü
5. cezveye kişi sayısı kadar su koy
6. kahveyi koy
7. karıştır
8. ocağa koy ve aç
9. kaynayana kadar bekle
10. ocağı kapat
11. fincanlara koy
12. servis et
13. bitir

BAŞLANGIÇ

bir çay bardağı getir

Su ısıtıcısına 300 ml su ekleyin

Su ısıtıcısı kendini kapatana kadar suyu kaynatın

Bardağın dibine yeni bir çay poşeti yerleştirin.

Su ısıtıcısından bardağa 200 ml kaynar su dökün

Çay poşetini 10 saniye güveçte bırakın

Çay poşetini 3 saniye karıştırmak için metal bir kaşık kullanın

Kaşığı kullanarak çay poşetini fincandan çıkarın

Çay poşetini çöpe atın

SON

veya çay demleme algoritması yapalım

hazır meyve suları fabrikada nasıl hazırlanıyor onları düşünelim

e mail gir denildiğinde önce formata uygun mu diye ona bakar, kaç harfli, ortada @ işareti var mı……

Her problemi neden yazdığımızı, algoritmayı ne için yazdığımızı tanımlamayı ihmal etmezsek daha sonra dönüp baktığımızda bu programı ne için yazdığımızı bulmak için zaman kaybetmemiş oluruz.

Whideboard

Bu sayıları sırala dediğimizde…. 75 51 28 14 1 99 12 36

Burada 12 nerede dediğimizde hemen bulabiliyoruz ama bilgisayar böyle değil. İnsan beyni çok hızlı, gördüğünü çok hızlı algılayabiliyor.

Ama bizimde zaafımız nedir, biz çok büyük işlemleri uzun sürelerde yapamıyoruz. Burada da bilgisayarın avantajı var. Biz ne yapacağız bizim mantığımızı bilmek, bilgisayarın düşünce mantığını bilmek ve problemleri ona göre adapte etmek

Bilgisayarlar sıralamayı nasıl yapıyor ?

Sort

1 2 3 4 5 6 7 8 9 10

Sort 2

6 4 7 10 3 1 5 8 2 9

Ben nasıl sıralama yaptıracağım bir karar vermem gerekiyor, soldan sağa karşılaştırmalı sıralama mı yaptırayım veya en küçüğü bulup ta en büyüğe devam mı et diyeceğim.

Ben bilgisayarı yan yana karşılaştırarak gidiyorum.

1 2 3 4 5 6 7 8 9 10-----bilgisayar ancak 10 defa karşılaştırdıktan sonra bana cevap veriyor.

Daha önce sort algoritması yazmanız gerekiyordu ama artık sizin bir sort algoritması yazmanıza gerek kalmayacak çünkü bunların kütüphaneleri var biz bunları kullanacağız.

\*\* 13 ü bulmak için artadan giriyor sağda mı solda mı ona bakıyor öbür tarafı atıyor kalan sayıların ortasından dalıyor yine bakıyor………

Sizden bu hesaplamaları yapmanız istenmeyecek bunların nasıl kullanılacağını bilmeniz yeter, hangi hesaplamalarda hangi teknikleri kullanacağınıza bakıp bilmeniz ve kullanmanız gerekli.

Selection sort----sayı gurubu içerisinden en hafif olanı bul ve başa yaz.

Quick sort----selekşin dan çok daha hızlıdır, genelde büyük listeler için kullanılır, bilinen en iyi yöntemlerden biridir.

Rastgele bir obje seç ve terazinin bir tarafına koy ve kalanların her birini bununla kıyasla hafif olanları sola ağır olanları sağa koy diyor. Ve bunu kalan guruplar içerisinde tek bir obje kalmayıncaya kadar devam ettir diyor.

Örneği yap

diyagram içeren bir resim

Açıklama otomatik olarak oluşturuldu

İnserting sort tekniği var ne diyor-----her bir objeyi tek tek sıralanmamış guruptan al ve onu doğru yerine koy.

7 ile başlayalım sonra rastgele bir sayı daha aldım

Bubble sort---- listenin üzerinden tekrar tekrar tekrar geçmek üzerine kurulmuş. Yan yana olanları tek tek karşılaştırıp dizmek şeklinde oluyor. Sıralama bitince de algoritmamın çıkış noktası oluyor.

Bu method etkin değildir ama anlaması ve kullanması basittir.

Merge sort----önce listeye rastgele iki eşit parçaya bölmemiz gerekiyor. Her bu iki eşit parça sıralanıyor ve bu listeler birleştiriliyor.

Evet Computişinıl Tiking in her alanda kullanılabilen bir consept olduğunu ama bizim bilgisayara bir problem çözdürmek istiyorsak bunun bir zorunluluk olduğunu görmüş olduk.

İnsanın sayıya bakınca tek seferde sıralayıp çözebiliyorken, bilgisayar bunun için 20 satırlık işlem yaptığını bilmiş olacağız. Bunları bilgisayara biz yaptıracağız, biz nasıl tanımlarsak o şekilde problemi çözecektir.