



PROGRAMLAMAYA GİRİŞ VE ALGORİTMA

Öğr.Gör.Erkan HÜRNALI

Öğrenme Hedefleri

- Programlama Sanatını Tanıyabilecek,
- Algoritma nedir? Tanımlayabilecek,
- Problem nedir? Tanımlayabilecek,
- Çevrenizdeki problemlerin farkında olabilecek,
- Problemlerin çözüm aşamalarını içeren algoritma adımlarını oluşturabilecek
- Algoritma adımlarının taşınması gereken özellikleri bileceksiniz.

Programlama Sanatı



sanat

ad

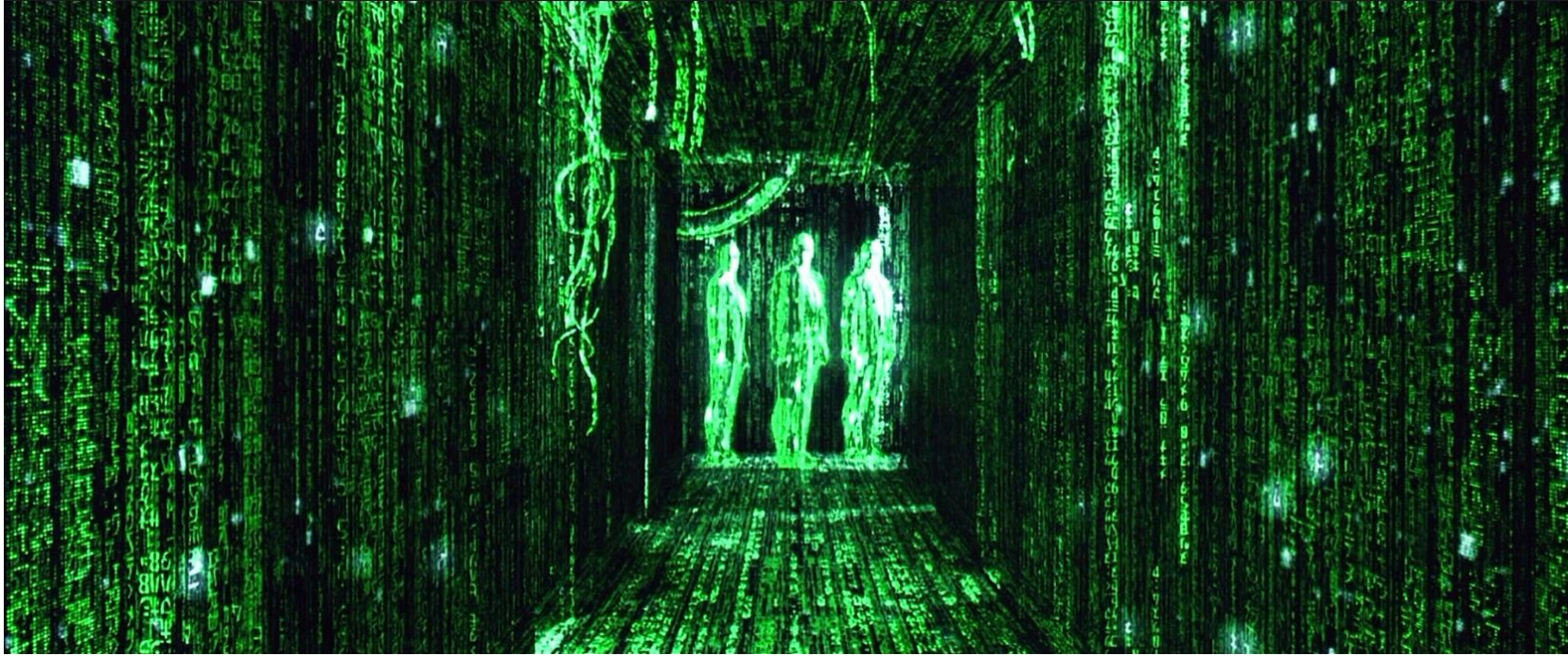
1. bir duygunun, tasarımın, güzelliğin vb. dışavurumunda, anlatımında kullanılan yöntemlerin tümü.
2. bu yöntemlerle ortaya konulan üstün yaratıcılık.
3. belli bir uygarlığın, belli bir dönemin anlayış ve beğeni ölçülerine uygun yaratılmış anlatım.
"Türk sanatı, antik sanat"
4. bir şeyi yapmakta gösterilen ustalık.
"Konuşma sanatını bilmiyoruz"
5. bir meslekte uyulması gereken kuralların tümü.
"Öğretmenlik sanatı"
6. kimi zaman zanaat yerine kullanılır.

Köken

Arapça

En genel anlamıyla **yaratıcılığın** ve **hayal gücünün** ifadesi olarak anlaşılır.

Programcı Gözüyle Bakmak



Algoritma Geliştirme

- Problem Nedir?
- Algoritma Nedir?
- Algoritma Adımları
- Algoritmaların Taşınması Gereken Özellikler
- Bilgisayar Programı Nedir?
- Algoritma Örnekleri
- Algoritma Çalıştırma

Problem Nedir?

Problem, çözümlmesi gereken sorun ya da aşılması gereken engel anlamına gelir.

Günlük hayatta sık sık problemlerle karşılaşırız.



Karşılaştığınız bir problemi çözmek için ne yaparsınız?



Bir Problemin Çözümü İçin...

Problemi
iyi
anlamak

Ve sonucun
doğruluğunu
kontrol etmek

Kısa ve
anlaşılır
biçimde
çözmek

Problem Çözme

Günlük yaşamda karşılaştığımız problemleri bilerek veya farkında olmadan adım adım çözmeye çalışırız.

Örneğin yazı yazarken kaleminizin ucu kırıldığında şu adımları takip ederek bu sorunu çözersiniz.



PROBLEM

1. Kalem tıraşı çıkar.
2. Kalemi al.
3. Çöp kovasının yanına git.
4. Kalemin ucunu aç.
5. Sırana geri dön.
6. Yazmaya devam et.

Peki Ya Bilgisayarlar?

Bilgisayarlar da problemleri tıpkı bizler gibi çözmeye çalışır. Kullanıcı tarafından kendisine verilen komutları adım adım uygulayarak problemin çözümüne ulaşır.

Kullandığımız yazılımların tamamı «kod» adı verilen bilgisayarın anlayacağı dilde yazılmış özel komutlardan oluşur.

Bu kodlar bilgisayar yazılımcıları tarafından yazılır.

```
author mino-
This class will use the ghost...
to insert a collection of particles which
have no net charge. This is used to calcul
chemical potential and activity coefficient

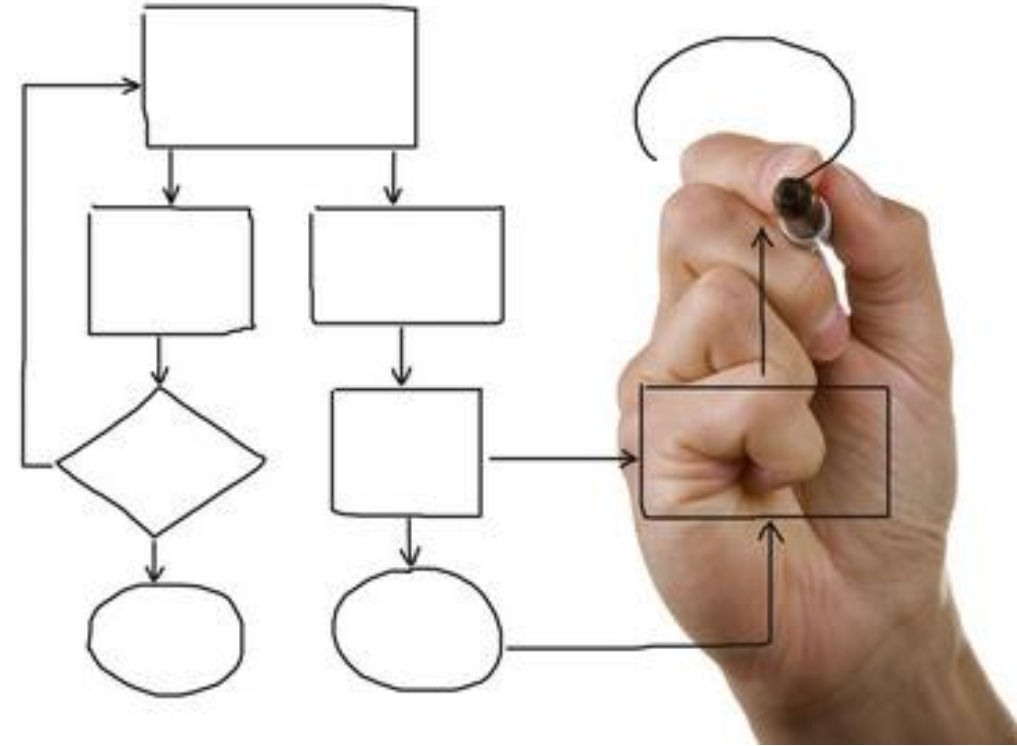
class widom : public analysis {
private:
    average<double> expsum; ///< Average of 1
protected:
    int ghostin;
    long long int cnt;
    vector<particle> g;
public:
    widom(int n=10);
    string info();
    void add(particle);
    void add(container &);
    void insert(container &, energybase &
    void check(checkValue &);
    ~widom() { return exp(muex());
    }
};
```

Kodlamadan Önce...

Kodlamaya başlamadan önce oluşturacağımız yazılımın adım adım ne yapacağını tasarlamamız gerekir.

İşte açık ve net ifadelerle problemin adım adım çözümünü gösteren bu taslağa «algoritma» adı verilir.

Programlamanın ilk adımı algoritma oluşturmaktır.

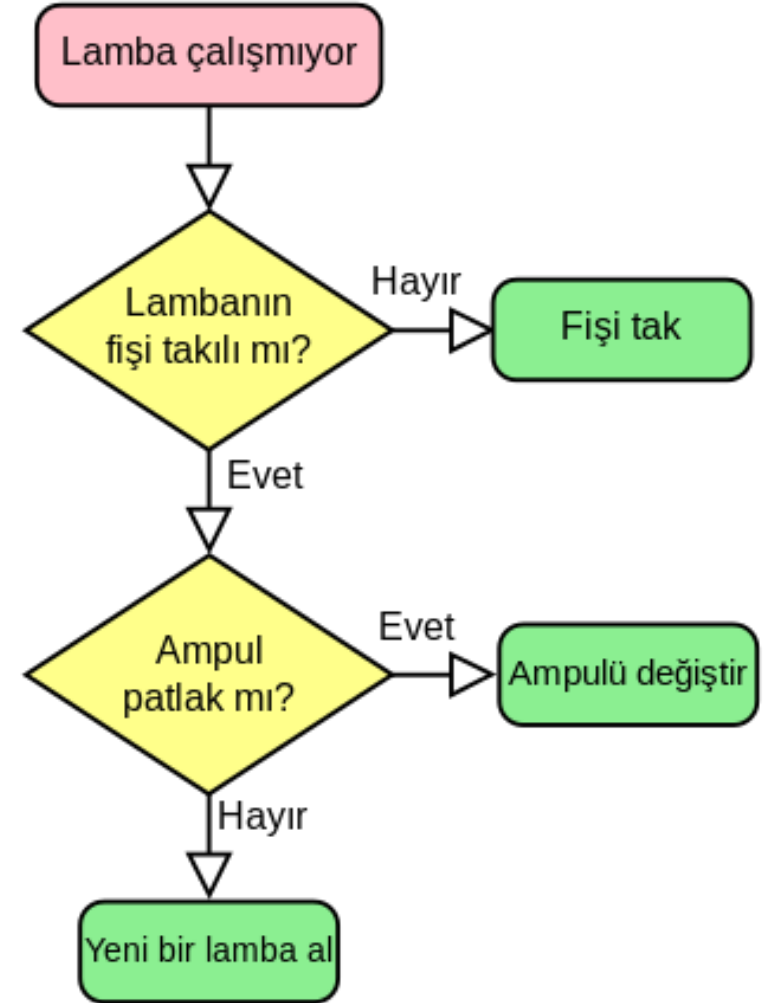


Algoritma

Bir problemin çözümünde izlenecek yol anlamına gelir ve problemin çözümünün adımlar halinde yazılmasıyla oluşturulur.

Algoritma basamaklarının bir başlangıcı ve sonu bulunur.

Her adımda yapılacak işlem açıkça belirtilir.



Algoritma Nedir?

- Algoritma en yalın tanımıyla bir **problemin çözüm aşamalarıdır.**
- Diğer bir ifadeyle problemi çözmek için **tespit edilen işlem basamaklarıdır.**

Örnek Algoritma

Şimdi basit bir problemin çözümünü gösteren bir algoritma hazırlayalım.

Ayran yapıp bardağa dolduralım.

- Adım 1: Başla
- Adım 2: Yoğurdu kaba koy.
- Adım 3: Su ekle.
- Adım 4: Çırp.
- Adım 5: Tuz koy.
- Adım 6: Bardağa doldur.
- Adım 7: Bitir.



Örnek Algoritma - 2

Arabayı çalıştırıp yola çikalım.



- Adım 1: Başla
- Adım 2: Sürücü koltuğuna geç.
- Adım 3: Emniyet kemerini tak.
- Adım 4: Aynaları kontrol et.
- Adım 5: Anahtarı tak.
- Adım 6: Kontakı çevir.
- Adım 7: El frenini indir.
- Adım 8: Vitese geç.
- Adım 9: Gaza bas.
- Adım 10: Bitir.



Problem: Okula Varmak

- Uyanmak (Uyanmadan okula gitmemeliyiz. Gidebilir miyiz? Evet, gidebiliriz fakat bu istenmeyen bir durumdur.)
- Elimizi yüzümüzü yıkamak. (Aynı şekilde elimizi yüzümüzü yıkamadan da okula gidebiliriz fakat bu da istenmeyen bir durumdur.)
- Kahvaltı yapmak (Yine aynı şekilde kahvaltı günün en önemli öğünlerindedir...)
- Hazırlık (...)
- Ulaşım araçlarını kullanmak (...)
- Okula varış (...**ve problem çözülmüş oldu**)

Problem Nedir?

- Gerçek hayatta problem, bir işlemin, otomasyonun ya da bilimsel hesaplamanın bilgisayarla çözülmesi fikrinin ortaya çıkmasıdır.
- Bu tip fikirlerde insanların bu sorunları beyinle çözmeleri ya imkânsızdır ya da çok zor ve zaman alıcıdır.
- Bu tip bir sorunu bilgisayarla çözebilme fikrinin ortaya çıkması bir bilgisayar probleminin ortaya çıkmasına neden olmuştur.
- Bazen de bir işletme veya yönetimin otomasyonunu sağlamak amacı ile bu tip problemler tanımlanır.

Neden Algoritma Kullanıyoruz?



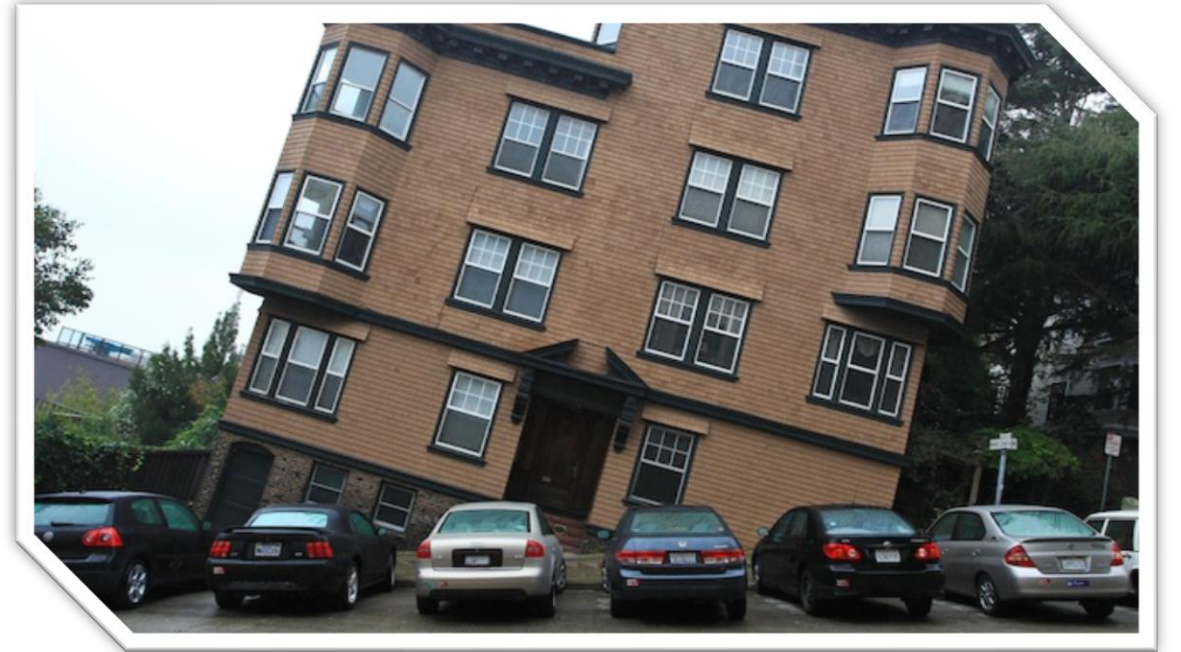
Sizce kodlamaya başlamadan önce niçin algoritma hazırlıyoruz?

**Gerekli tüm bilgi ve birikime sahipsiniz ve sizden bir bina yapmanız isteniyor.
Yapacağınız ilk iş ne olurdu?**



Neden Algoritma Kullanıyoruz?

Oluşturacağımız yazılımın kusursuz olması için öncelikle her adımını gösteren planını, yani algoritmasını hazırlamalıyız.



Algoritma Adımları

- Uyan
- Uyanıp-uyanmadığını kontrol et
- Eğer uyanmadıysan başa dön, uyandıysan devam et...

Algoritma Adımları

Programcılık dilinde numaralandırılmış olan bu işlem basamaklarının her birine bir “algoritma adımı” denmektedir.

Yukarıdaki aşamayı algoritma adımları ile ifade edecek olursak aşağıdaki gibi olur:

1. Uyan
2. Uyanıp-uyanmadığını kontrol et
3. Eğer uyanmadıysan 1. Adıma geri dön, uyandıysan 4.Adımdan devam et
4. Elini-yüzünü yıka...

İyinin de iyisi var 😊

1. Sayıları oku
2. Sayıların toplamlarını hesapla
3. Toplamlarını ekrana yaz



1. Başla
2. Sayıları oku
3. Sayıların toplamlarını hesapla
4. Toplamlarını ekrana yaz
5. Dur



1. Başla
2. A sayısını oku
3. B sayısını oku
4. $C = A + B$
5. C sayısını yaz (ekrana yaz)
6. Dur



1. Başla
2. A ve B sayılarını oku
3. A ve B sayılarının toplamlarını hesapla
4. Toplamlarını ekrana yaz
5. Dur

Algoritmaların Taşınması Gereken Özellikler

- Her adım son derece belirleyici olmalıdır ve aynı zamanda hiç bir şansa bağlı olmamalıdır (yukarıdaki örnekte uyanmak adımının geliştirilerek şansa bırakılmadığı gibi)
- Belirli bir sayıda adım sonunda algoritma sonlanmalıdır (sonsuz bir döngüye girmemelidir)
- Algoritmalar karşılaşılabilecek tüm ihtimalleri ele alabilecek kadar genel olmalıdır.