

# Paralel Bilgisayarlar Projesi

## SATisfiability NP-Tam (NP-Complete) Problemi

**Polinom problemler:** işlenen elemanın bir polinomu ile sınırlandırılırsa polinom fonksiyonlardır. Örnek: Bubblesort algoritması  $O(n^2)$  dir.

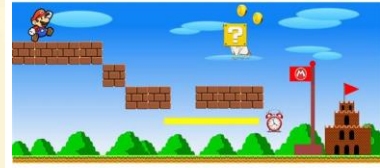
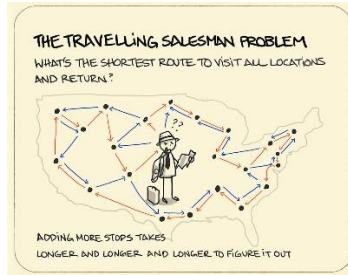
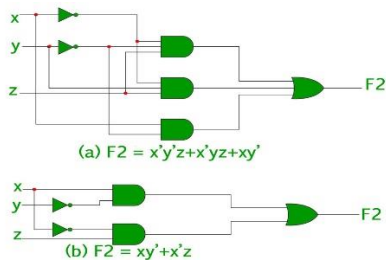
**Non-Polynomial problemler:** Çözümü Polinom zamanda bulunamayan ama en kötü ihtimalle üssel olarak çözülebilen problemlerdir.

**NP-Tam problemler:**

**Graf Algoritmaları:** Klik(clique), graf örtüsü (vertex cover).

**Optimizasyon:** Gezen satıcı problem.

**Devre optimizasyonu:**



**Tanımlar:**

**Değişken (önerme):** cevabı evet veya hayır olan ifade.

Güneş dünyadan büyüktür: evet

Hava bulutludur: evet/hayır

X

**Operatörler/işlemler**

- 1- a'nın değili:  $\text{NOT}(a) = \neg a, \sim a, \neg a, a'$
- 2- VEYA işlemi:  $|, +$ ,
- 3- VE işlemi :  $\&, \cdot, \cdot$ ,
- 4- XOR işlemi:  $\oplus$
- 5- ...

**Literal:** bir önermeyi/değişkeni veya tersini göstermektedir.

**a bir önerme ise, a'nın değili:**  $\text{NOT}(a) = \neg a, \sim a, \neg a, a'$

**a:** doğru ise  $\neg a$  yanlıştır,

**$\neg a$ :** doğru ise a yanlıştır.

**Cümle:** birden fazla literalin VEYA işlemi ile gruplanması;

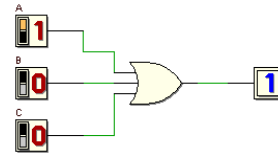
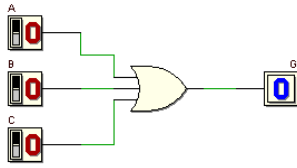
$$(a \text{ VEYA } b \text{ VEYA } c) = (a \mid b \mid c) = (a, b, c)$$

**Conjunction Normal form:** Birden fazla cümlenin AND işlemiyle bağlanması;

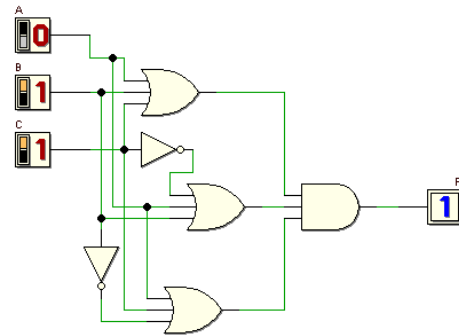
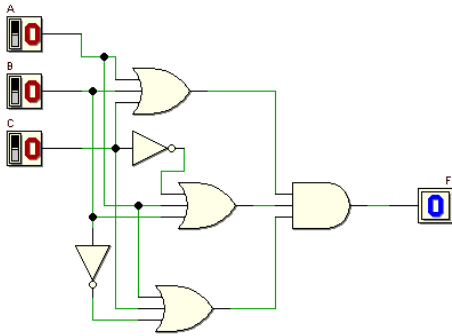
$$\begin{aligned} & (a \text{ VEYA } b \text{ VEYA } c) \& (a \text{ VEYA } b \text{ VEYA } \neg c) \& (a \text{ VEYA } \neg b \text{ VEYA } c) \\ & = (a \mid b \mid c) \& (a \mid b \mid \neg c) \& (a \mid \neg b \mid c) \\ & = (a, b, c) (a, b, \neg c) (a, \neg b, c) \end{aligned}$$

**Not:** Her Boolean ifade CNF formda ifade edilebilir.

$$G = (a, b, c)$$



$$F = (a, b, c) (a, b, \neg c) (a, \neg b, c)$$



## SAT Probleminin Tanımı

SAT probleminin aydınlatması için bkz. [stringfixer.com](http://stringfixer.com) , [frWiki](http://frWiki).

**SAT probleminin tanımı:** CNF formda verilen bir fonksiyonun çözümü var ise bir tanesini türet. Yok ise cevabı “yok” cevabını ver. Bu işlemi bütün problemler için polinom zamanda türeten bir algoritma bulunursa  $P=NP$  yoksa  $P \neq NP$ ’dir.

Aşağıdaki formülde “a” seçilirse bütün cümleler 1 üretir.

## Dosya format

Dimacs dosya formatında problemler, CNF (3CNF) olarak verilmiştir.

c Yorum satırı (comment)

c 3 değişken ve 4 cümle var

p cnf 3 4

1 2 3 0                      Yani    (1, 2, 3)

1 2 -3 0

1 -2 3 0

1 -2 -3 0

Sizden DIMACS formatında hazırlanan 10 tane 3-SAT problemin çözümü istenmektedir.

## Proje Kuralları:

- Gruplar min: 2, max:4 kişi (fazla veya az üyeli gruplara %15 kesinti yapılır)
- Raporun ilk sayfasında grup üyelerinin **numara** ve **isimleri** açık bir şekilde (1/2 öğretim bilgisi dâhil) yazılmalıdır.
- Her gruptan **bir arkadaş**, grubun hazırladığı raporu (pdf formatında) ve program kodlarını (ve exe dosyasını) iki dosya olarak sisteme (Moodle) yükleyecektir.
  - Dosya isimleri grupta bulunan bir arkadaşın bilgisi ile (no\_ad\_soyad.pdf),
  - (kodlar+exe) dosyaları grupta bulunan bir arkadaşın (no\_ad\_soyad.zip)
  - Videolar drive’dan (Google, onedrive vb.) **paylaşılacaksa erişim izni verilmeli...**
- Proje sonuçları raporda (video +15p) olarak açıklanmalıdır.
- Rapor minimum 10 sayfa olmalıdır.
- Projede elde edilen sonuçlar analitik bir şekilde açıklanmalıdır (Sadece ekran görüntüsü olmasın).
- Raporlar ciddi bir şekilde yazılmalıdır.
- **Proje dosyaları mailden kabul edilmemektedir.**
- **Raporu düzgün olmayan grubun projeden 0 alır. Raporlara ek süre verilir ve geç yükleyen grubun notu %15 kesilir.**
- Raporda size verilen 10 tane CNF dosyasının çözümü istenmektedir. Çözümü bulunmayan problemler tabloda (UNSAT) olarak kodlanmalıdır. Eğer problem SAT ise (varsa) ilk 20 değişkenin değeri tablo halinde verilmelidir.

#### Örnek tablo:

No	Pr1	Pr2	Pr3	Pr4	...	Pr10
1	1	1		UNSAT		-1
2	-2	2				-2
3	3	-3				
4	4	-4				
...		...				
20						

- Hazırladığınız paralel uygulamada yük dağılımını ne şekilde düzenlediğinizi açıklayınız.
- Hazırlanan uygulamalarda elde edilen hızlanma/verimlilik değerlerini çıkartınız. Grafik ve tablolarla sonuçları karşılaştırınız.

#### Programlama Platformları:

- Java/Python/Diğer %25
- Web teknolojileri: PHP/Django %35
- OpenMP/pThread %30
- MPI/CUDA %35
- OpenCL %40
- Hadoop/Apache Spark %45
- Android/ %45

#### Nasıl yaparsınız?

- (%20) Öncelikle **kaba kuvvet yöntemi** ile seri bir şekilde çözün.
  - Bu durumda bütün değişken değerleri test edilir ( $2^n$ ).
  - Kombinasyonlardan birisi **problemi sağlar ise** bu kombinasyon çözüm olarak yazılır.
  - Hiçbir geçerli atama yoksa UNSAT (çözüm yoktur) yazılır.
  - $n > 20$  problemler zor olabilir.
- (%30) Paralel çözüm-1: Seri işlemin paralel yapılması.
  - Bütün kombinasyonlar multi-thread ile test edilir ( $2^n/p$ ).  $p$  tane thread varsa...
  - Kombinasyonlardan birisi **problemi sağlar ise** bu kombinasyon çözüm olarak yazılır program sonlandırılır.
  - Hiçbir geçerli atama yoksa UNSAT (çözüm yoktur) yazılır.
- (%50) Paralel çözüm-2: Resolution işlemi (DPLL algoritması) ile
  - Resolution( (a, b, c) (a, b, -c) ) = (a, b)
  - Resolution( (a, b, c) (-a, b, d) ) = (b, c, d)
  - Resolution( (a, b, c) (-c, d, e) ) = (a, b, c, d) şeklinde türer
  - Resolution( (a, b, c) (-a, -b, e) ) = **HATA**
  - **Maksimum 6'lı cümle yaparsınız...**
  - (a, b, c) (-c, d, e) (-e, f, g) = (a, b, d, f, g) 5 li cümle...
  - (a, b, c) (-c, d, e) (-e, f, g) (-g, h, i) = (a, b, d, f, h, i) 6 li cümle...

**Başarılar...**