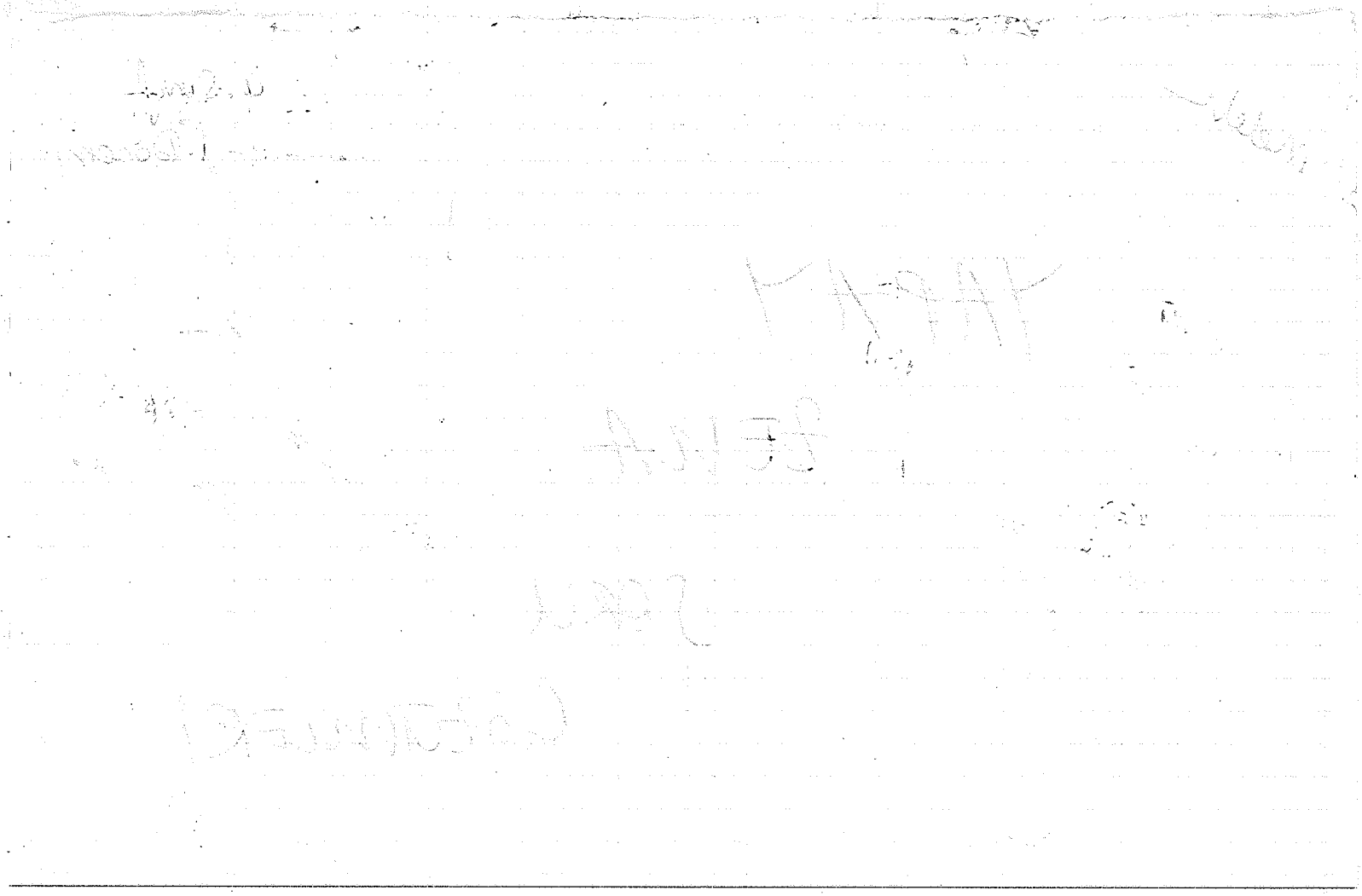
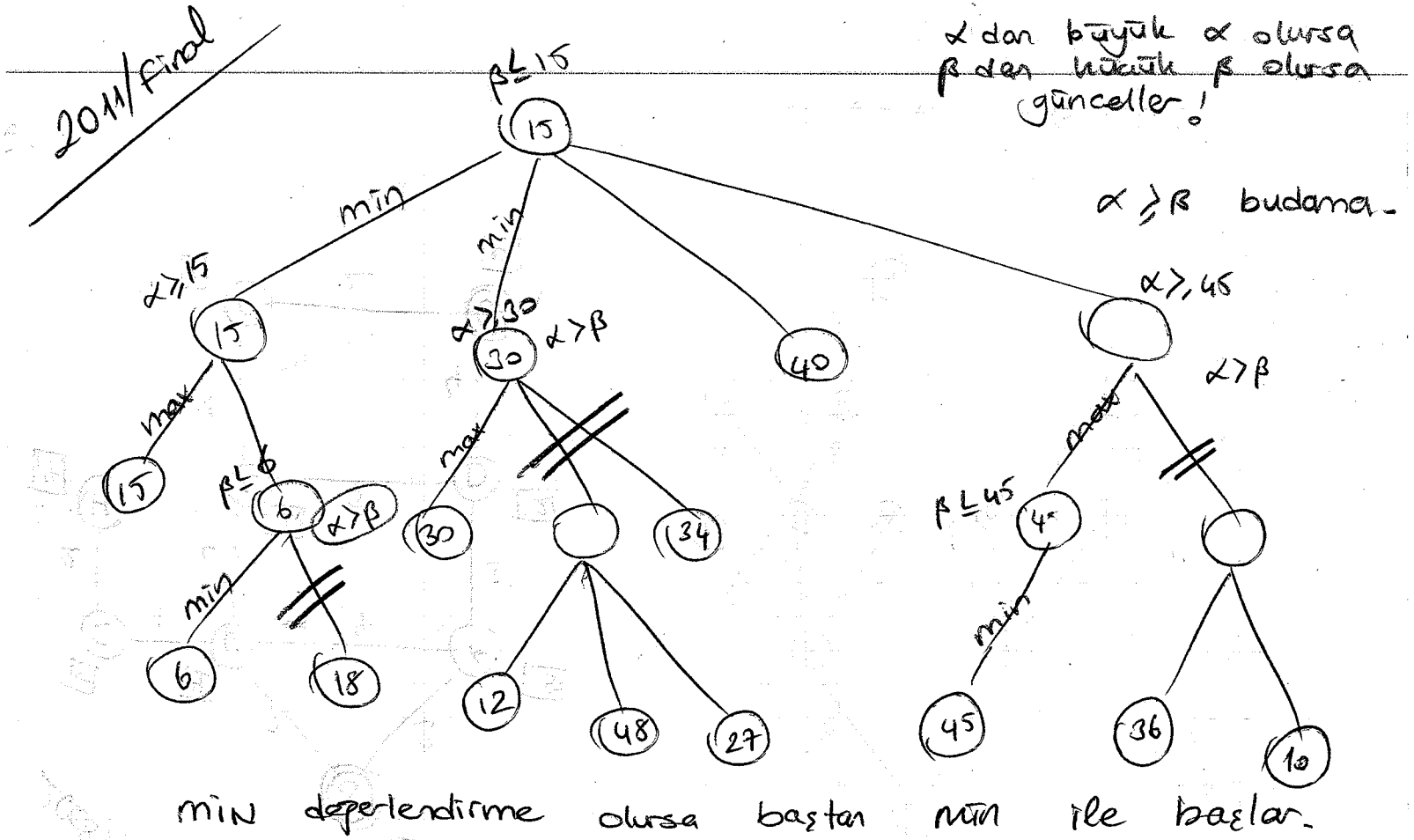
Melek

4. Sınıf
1. Dönem

YAPAY ZEKA SORU GÖZÜMLERİ

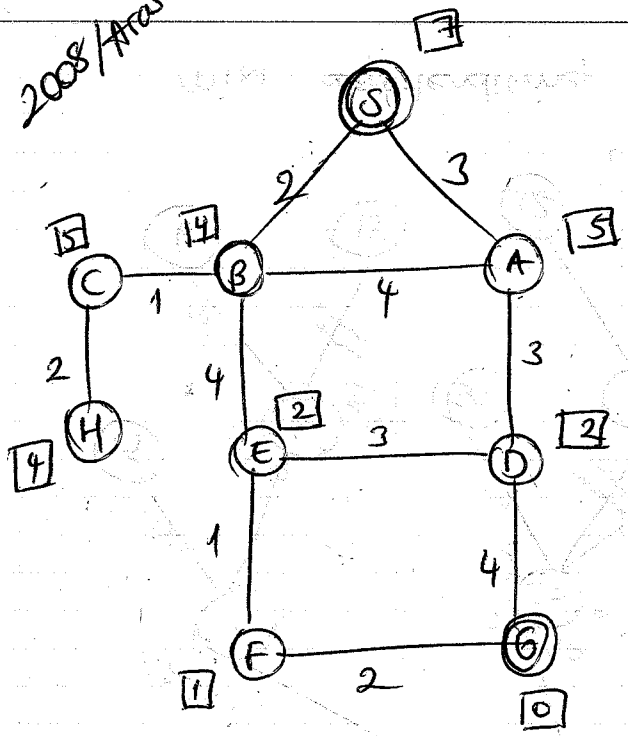


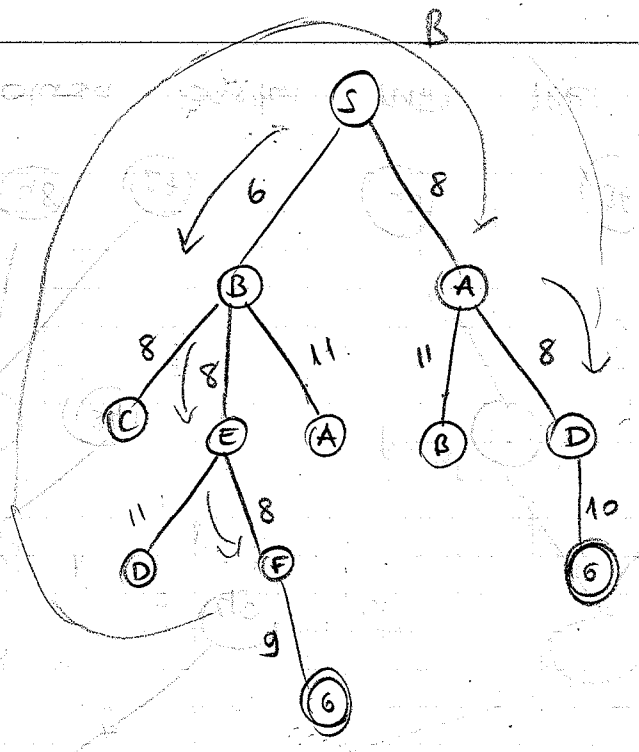
2011/Final



MIN değerlendirme olursa baştan MIN ile başlar.

2008 / Απασινάκη





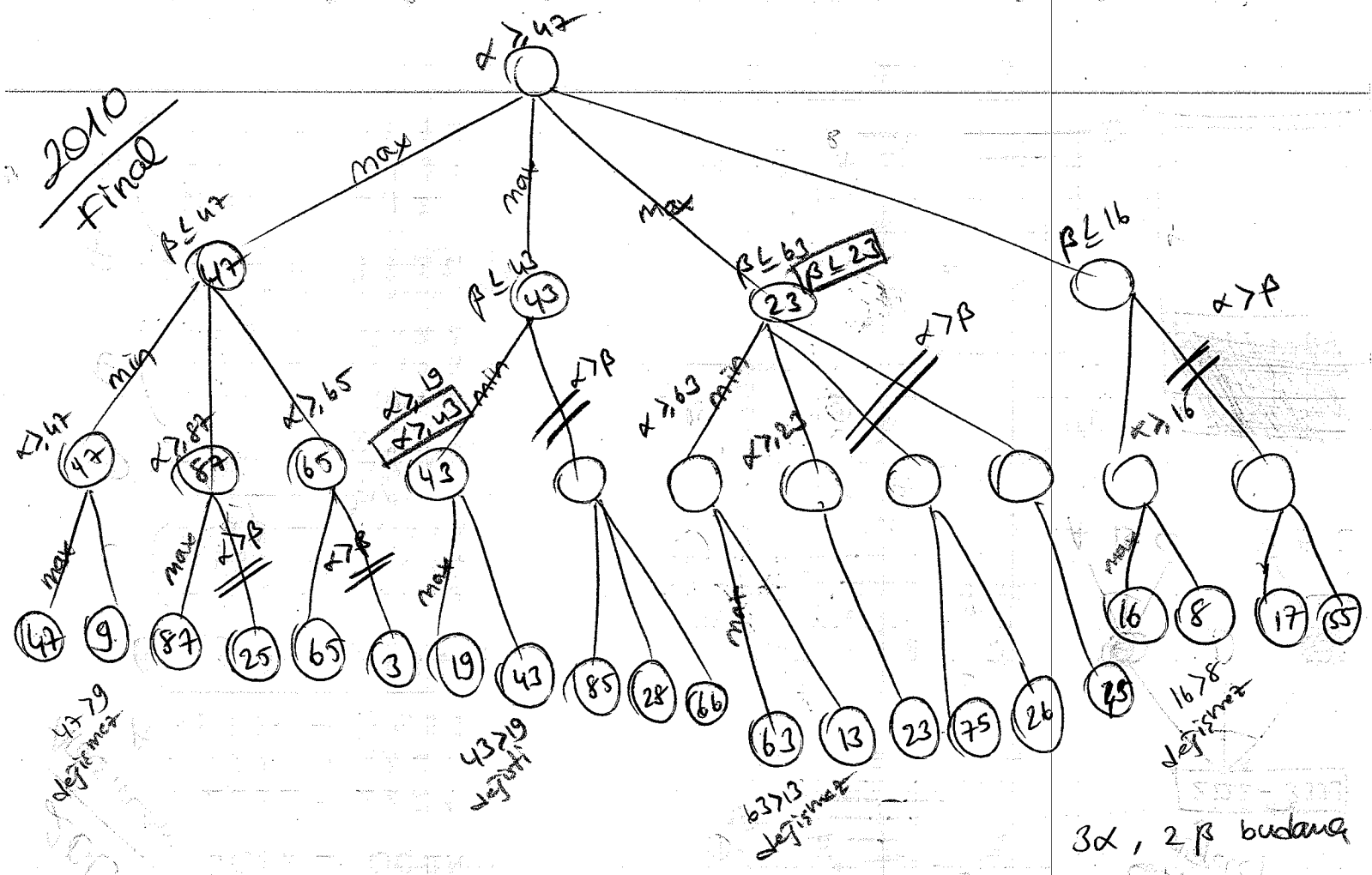
$fs \leq 8$

$fs \leq 9$

$fs \leq 10$

S - B - E - F - G $\rightarrow$ 9

2010
final


α ≥ 47
max
β ≤ 47
47
α ≥ 47
min
47
9
α ≥ 87
87
25
α ≥ 65
65
3
β ≤ 43
43
α ≥ 19
α ≥ 43
43
19
43
α < β
85
28
66
β ≤ 63
β ≤ 23
23
α ≥ 63
63
α ≥ 23
13
23
75
26
α < β
β ≤ 16
α ≥ 16
16
8
α < β
17
55
25
47 > 9 dejtsmet
43 > 19 dejtsmet
63 > 13 dejtsmet
16 > 8 dejtsmet
3α, 2β budang

2013 Atasınav

DCBA – DCBA

A
- 2222 – 2221
- 2221 – 2231
- 2231 – 2233

B
- 2233 – 2133

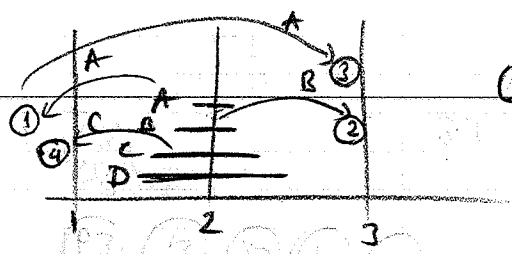
C
- 2133 – 2132
- 2132 – 2112
- 2112 – 2111

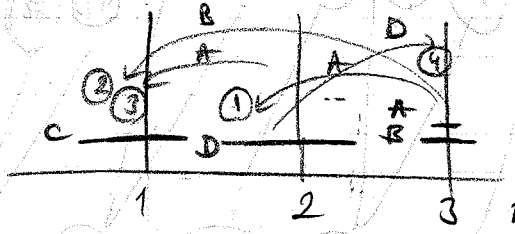
D
- 2111 – 3111

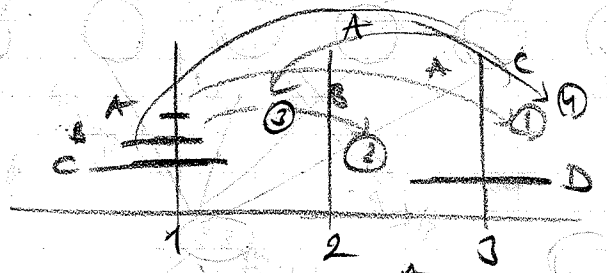
E
- 3111 – 3113
- 3113 – 3123
- 3123 – 3122

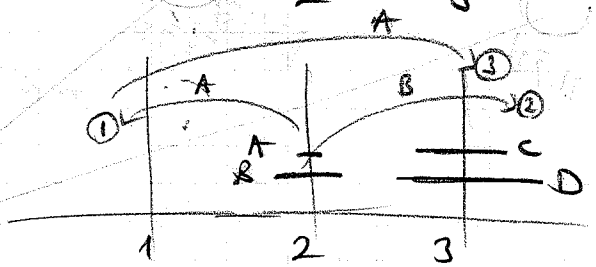
F
- 3122 – 3322

G
- 3322 – 3321
- 3321 – 3331
- 3331 – 3333

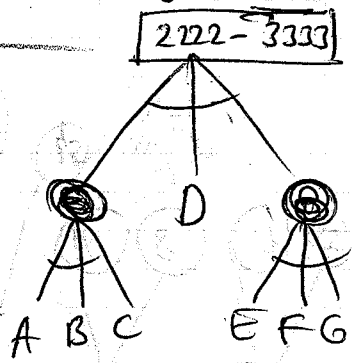


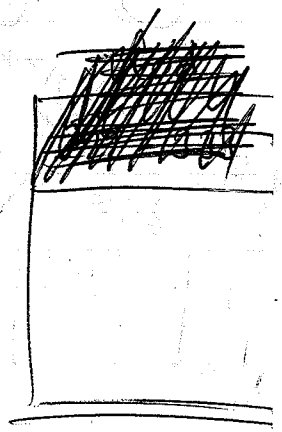




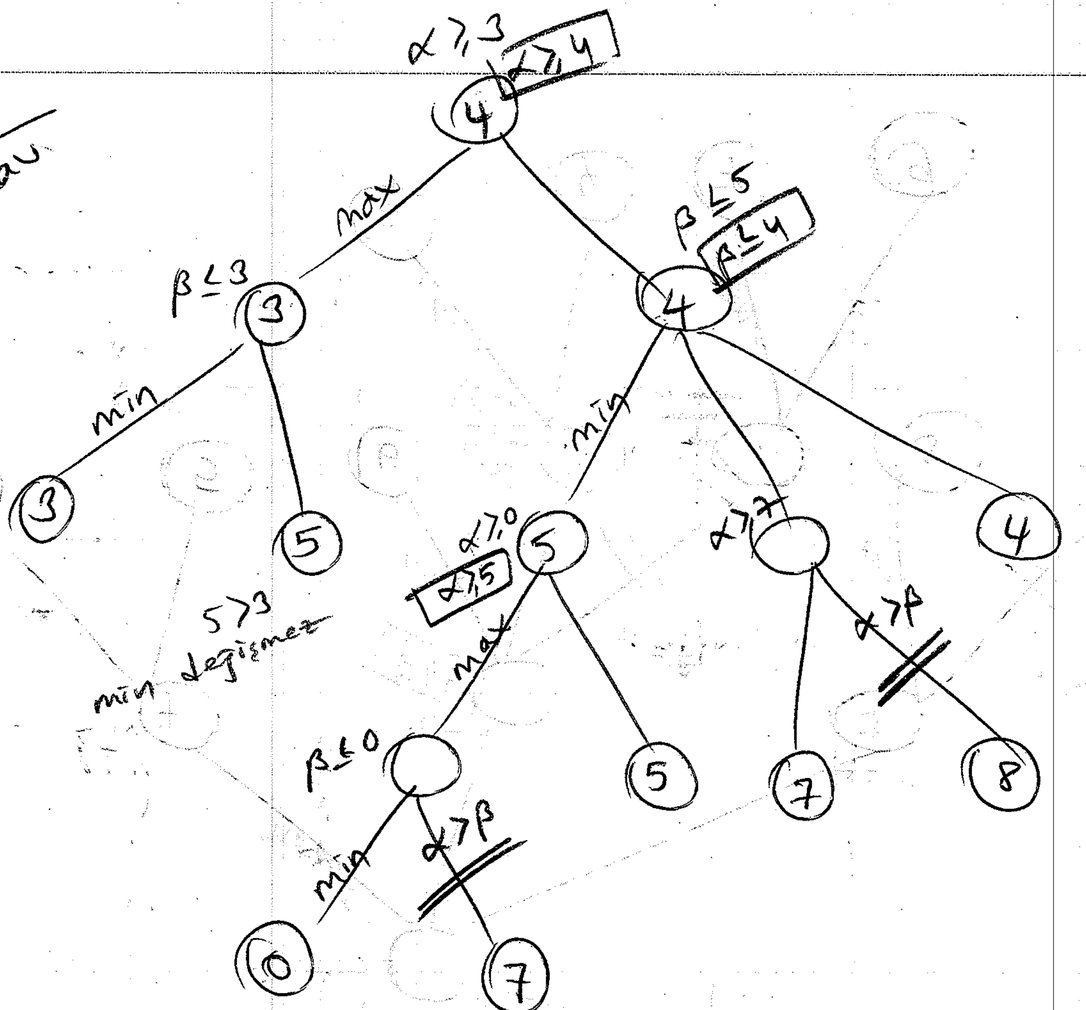


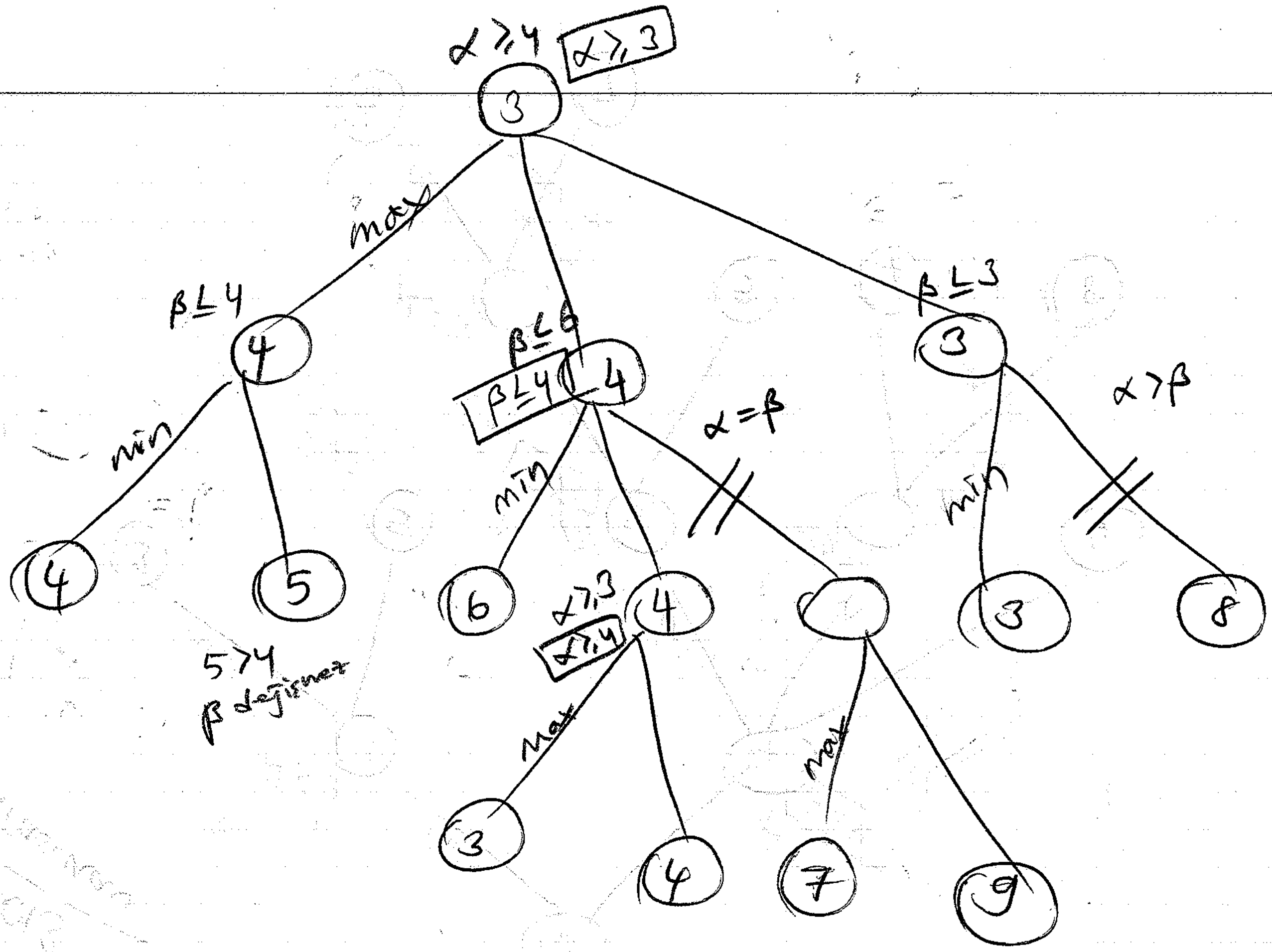
Çözüm ağacı

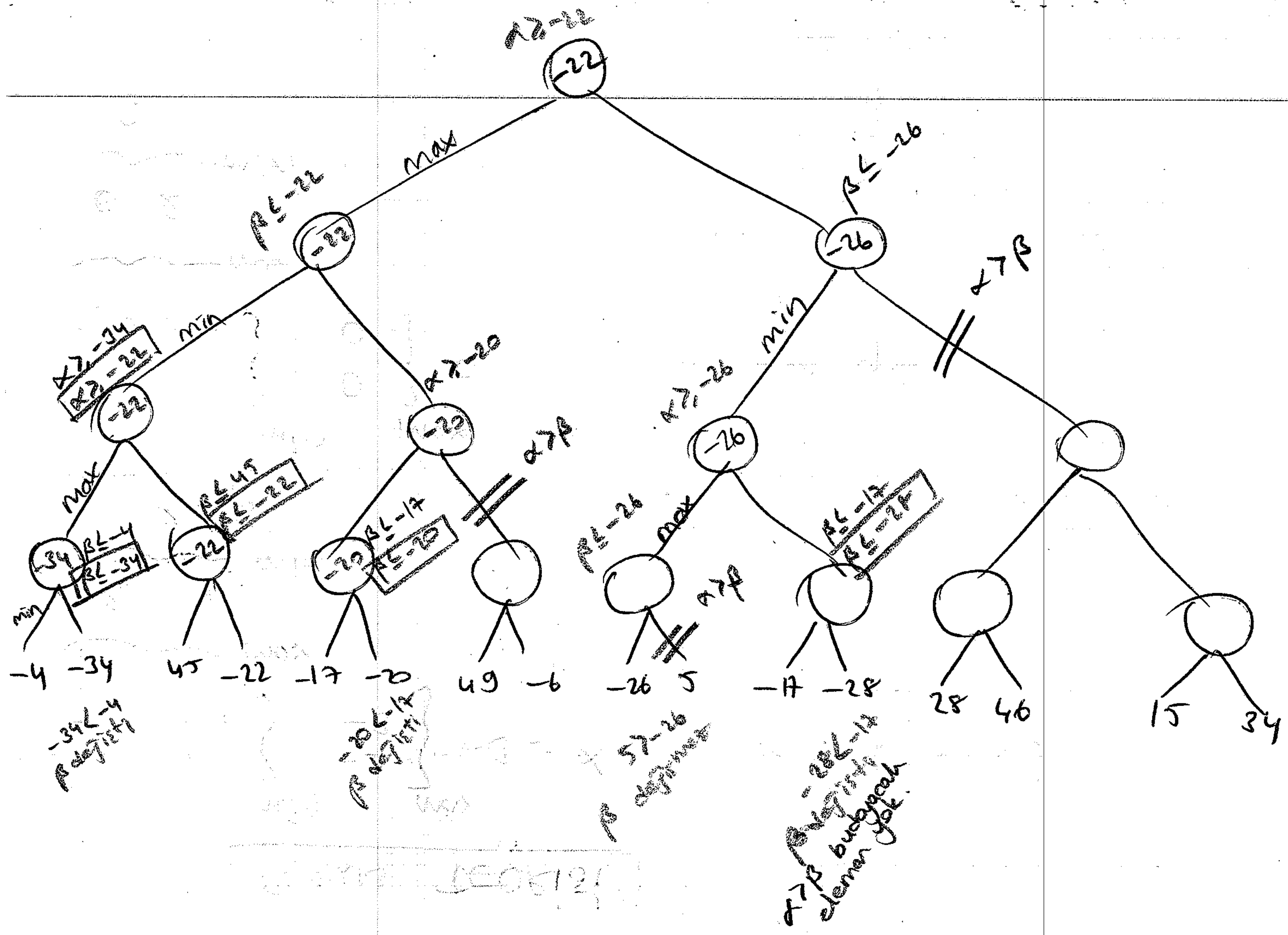




2013
Arasınav




α ≥ 4
α ≥ 3
3
max
β ≤ 4
4
β ≤ 6
β ≤ 4
4
β ≤ 3
3
min
α = β
α > β
4
5
6
α ≥ 3
α ≥ 4
4
3
8
5 > 4
β değişmez
max
3
4
7
9



α ≥ -22
-22
max
β ≤ -22
-22
β ≤ -26
-26
α > β
min
α ≥ -34
α ≥ -22
-22
α ≥ -20
-20
α > β
min
α ≥ -26
-26
max
β ≤ -4
β ≤ -34
-34
β ≤ 45
β ≤ -22
-22
β ≤ -17
β ≤ -20
-20
β ≤ -26
max
α > β
β ≤ -17
β ≤ -28
min
-4 -34 45 -22 -17 -20 49 -6 -26 5 -17 -28 28 46 15 34
-34 < -4
β değişti
-20 < -17
β değişti
5 > -26
β değişmez
-28 < -17
β değişti
α > β budayacak eleman yok.

OYUN TEORİSİ

$$\begin{pmatrix} 2 & -3 \\ -3 & 4 \end{pmatrix} \begin{matrix} \text{min} \\ -3 \\ -3 \end{matrix} \begin{matrix} \text{max} \\ \\ \end{matrix} \quad -3 = \alpha \qquad \mu \in [-3, 2]$$

max

2 4

min

$2 = \beta$

$$\begin{pmatrix} 0 & 1 \\ 0 & 2 \end{pmatrix} \begin{matrix} \text{min} \\ 0 \\ 0 \end{matrix} \begin{matrix} \text{max} \\ \\ \end{matrix} \quad 0$$

max

0 2

min

0

$q_1 = \dfrac{D - a_{12}}{a_{11} - a_{12}}$

a_{11} a_{12} a_{21} a_{22}

Kararan yok!

$$p_1 = \frac{a_{22} - a_{21}}{(a_{22} + a_{11}) - (a_{12} + a_{21})} = \frac{4+3}{6+(-6)} = \frac{7}{12}$$

$$p_2 = 1 - p_1 = \frac{5}{12}$$

$$D = 2 \cdot 7/12 + (-3)\, 5/12 = -1/12$$

2013 final

8 taş problemi karmaşıklığı $O(4^n)$ dir.

Bir basit sıralama problemi $O(n^2)$ ile sınırlıdır.

Günümüzün teknolojisi ile sıralama prb. çözülebilir problem boyu 1000. 8 taş prb. ise 100 ise 4096 luk güçlü bilgisayarlarda çözülebilecek problem boyu?

$$4096 = 4^6 = 64^2$$

$$4^{100} \cdot 4^6 = 4^{106} \qquad n = 106$$

$$(1000)^2 \cdot (64)^2 = (64000)^2 \longrightarrow 64000$$

Hanoi C kodu: (111-333)

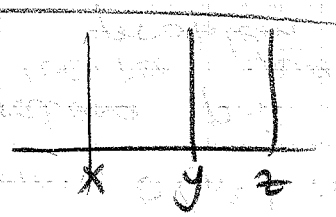
```
void hanoi (int n, char x, char y, char z)
{
  if (n==1)
      printf ("Diski x'den z'ye koy", x, z);
  else
  {
    hanoi (n-1, x, z, y);   // kaynak: x, hedef: z, ara: y
    hanoi (1, x, y, z);
    hanoi (n-1, y, x, z);
  }
}
```

$2^d - 1$

d

222-333

$n-1$	y	z	x
1	y	x	z
$n-1$	x	y	z



$3^3 = 27$ $2^3 - 1 = 7$

Bu kadar durumda çözülür

q^d mümkün durum sayısı

$N = 2^d - 1$ Geçişli durum sayısı

$N.2+1$ = çözüm sayısı

3	5	7
2	6	1
4	8	–

→

1	2	3
4	5	6
7	8	–

qoşum mümkünlüyü?

3 5 7 2 6 1 4 8

3 1 5 7 2 6 4 8

1 5 3 7 2 6 4 8

1 5 2 3 7 6 4 8

1 2 3 5 7 6 4 8

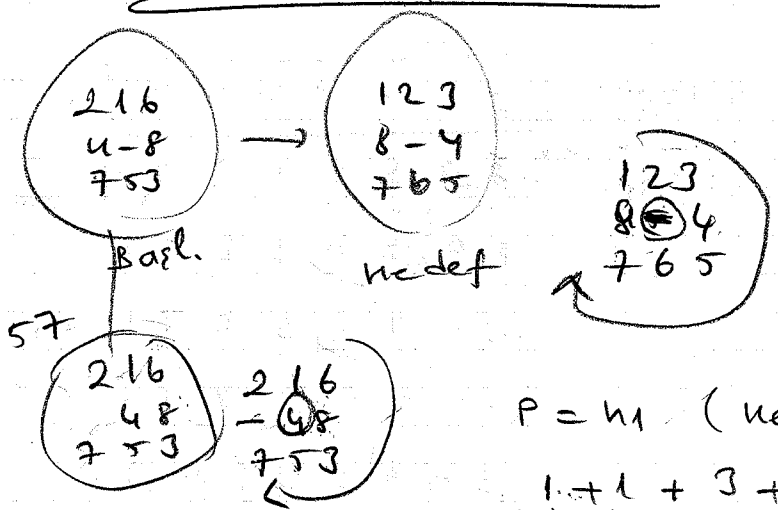
1 2 3 5 4 7 6 8

1 2 3 4 5 7 6 8

1 2 3 4 5 6 7 8 qoşum mümkündür.

Güçlendirilmiş Fork!

$h_3 = p + 35$ (h_1 + ceza)



$P = h_1$ (kendi yerine uzaklığı)

1 + 1 + 3 + 1 + 2 + 0 + 1 + 2
(2) (1) (6) (4) (F) (7) (5) (3) = 11

S (ceza) = 2 + 2 + 2 + 2 + 2 + 2 + 2 + 1 = 15 Sonraki yerin 2 ortada 1 yok

2 1 6 8 3 5 7 orta

$h_3 = \underbrace{11}_{h_1} + \underbrace{3.15}_{5} + \underbrace{1}_{\text{derinlik}} = 57$

Stern: IQ (Intelligence Quotient) – zeka bölümü kavramını önerdi.

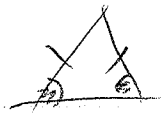
$$ZB = \frac{ZY.100}{Takvim\ Y.}$$

Turing: Sistemlerin zeki davranışlarının kontrolü

Sibernetik: İnsan ve hayvan davranışlarının taklidi.

Simon Newell'e göre Yapay Zeka;

- 10 yıl içinde bilg. satranç şampiyonu olacak +
- 10 yıl " bilg. yeni teoremler ispatlayacak +
- 10 " " bilg. psikolojik durum değerlendirmesi yapabilecek –



CUT !

if else gibi doğruysa devam eder. yanlışsa atla geçer.

```
p :- a, b, !, c.
p :- d, e, !.
p :- f.
```

$p = abc \vee \bar{b}de \vee \bar{b}\bar{e}f$

```
p :- a, b.
p :- c, !, d.
p :- e.
```

$ab \vee cd \vee \bar{c}e$

```
p :- a, d, !, b.
p :- c.
```

$adb \vee ad\bar{c}$

```
p :- a, !, b, c, d
p :- m.
```

$abcd \vee \bar{a}m$

```
p :- a, !, b.
p :- c.
```

$ab \vee \bar{a}c$

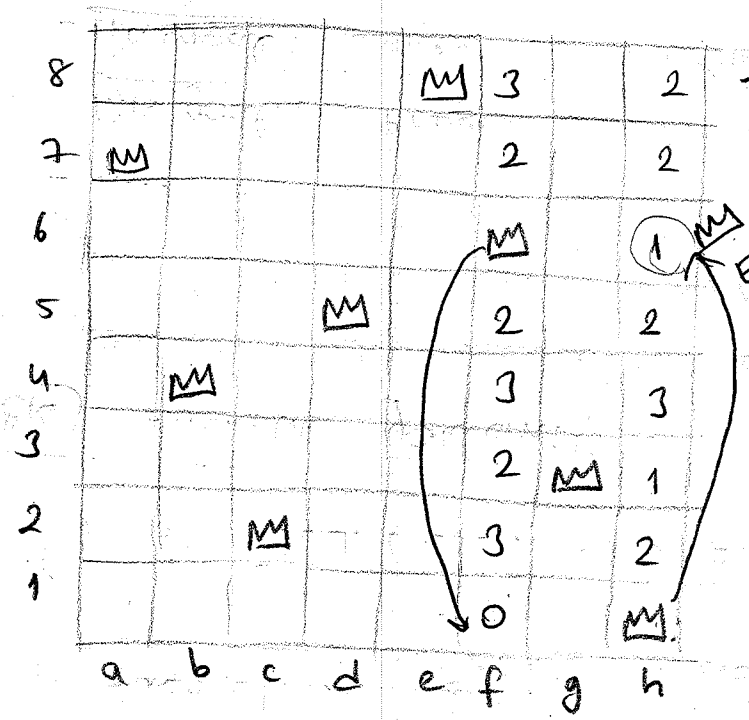
```
p :- a, !, b, c, d, !.
p :- m.
```

$abcd \vee \bar{a}m \vee \bar{d}m$

```
p :- a, !, b, !.
p :- c.
```

$ab \vee \bar{a}c \vee a\bar{b}c$.

Sezgisel Onarım Yöntemi :



Çözüm: [7, 4, 2, 5, 8, 1, 3, 6]

PROLOG

ÖRN: faktöriyel yüklemi?

```
fact(1,1).
fact(X,C) :- X>0, N=X-1, fact(N,AC), C=X*AC.
```

↳ 5! fact(5,C) :- 5>0 N=5-1=4 fact(4,AC) C=5*AC
(4! ↓, 4! ↓)

ÖRN: yatallık yüklemi?

```
yol(X,Y,Z) :- yatallık(X,Y,Z).   Kenar
yol(X,Y,Z) :- yatallık(X,N,Z1), yol(N,Y,Z2), Z=Z1+Z2.
```

Dış goal: Programa soru sorulması dir(X,Z).

İç goal: ogrenci(X), write(X).

Yeşil cut: Sadece işlemi hızlandırmak için. CUT kaldırılınca değişebilir sonuç.

```
f(x,0) :- x<3, !.
f(x,2) :- 3=<x, x<6, !.
f(x,4) :- 6=<x.
```

Kırmızı cut: CUT kaldırılırsa değişebilir olur.

```
f(x,0) :- x<3, !.
f(x,2) :- x<6, !.
f(x,4).
```

```
max(x,Y,x) :- x>=y, !.
max(x,Y,Y).
```

Listeye Üyelik:

uye(X, [X|_]). Listenin başında.

uye(X, [_|K]) :- uye(X,K). Listenin kuyruğunda.

→ uye(5, [2,3,5,7,9]) 5 liste üyesi mi? TRUE

→ uye(X, [2,3,5,7,9]) Liste hangi elemanlardan oluşur?

X=2 X=3 X=5 X=7 X=9

uye(X, [X|_]) :-!. CUT

uye(X, [_|K]) :- uye(X,K).

→ uye(X, [2,3,5,7,9]). X=2

CUTLI Listenin başında bulunca diğerlerine bakmaz.

Liste Uzunluğu:

uzunluk ([], ϕ)

uzunluk ([_|K] , L :- uzunluk (K , KU) , L = KU + 1

Başlalı eleman silinir. Baş eleman değiştirilir. Baş eleman boş olana kadar tekrarlanır. İlk silinen eklenir.

Listeye Eleman Ekle:

ekle (X , L , [X | L]). Liste başına eklene.

---

ekle (X , L , L) :- uye (X , L) , !. X, Listenin üyesiyse sonuç Listenin kendisi. (kırmızı CUT)

ekle (X , L , [X | L]).

Listeden Eleman Silme!

```
sil(X, [X|K], K) :- !.
sil(X, [B|K], [B|L1]) :- sil(X, K, L1).
```

Başta ise siler kuyruk döner CUT ile tek eleman siler.

İlk liste — son liste

Kuyruktan sil son liste döner.

```
sil(_, [], []).
sil(X, [X|K], M) :- !, sil(X, K, M).
sil(X, [B|K], [B|L]) :- sil(X, K, L).
```

Aynı değerli tüm veriler silinir.

```
sil(3, [3,2,3], Y), fail.
```

tüm sonuçları listeler

```
Y = [2,3]
Y = [3,2]
```

sil(X, [X,-|L], M)

Birini sil
birini atla-

sil(4, L, [1,2,3])

Silinmemiş hali:
→ 4 [1 2 3] → [1 2 4]
→ [1 4 2 3] → [1 2 3 4]

Liste Elemanlarının Yazdırılması:

```
writelist([]).
writelist([B|K]):- write(B), writelist(K).
```

Başı yazdır kuyruk boş olana kadar recursive işlem.

Listelerin Birleştirilmesi :

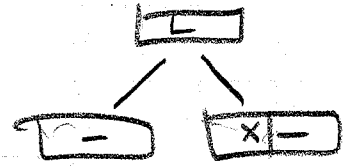

```
birl ( [], L, L).
birl ( [X|L1], L2, [X|L3] ) :- birl (L1,L2,L3).

birl ( x, [mayis | Y], [o,s,m,n, m,h,t, a,e,e,h, a] ).
```

X: mayıs öncesi — Y: mayıs sonrası,

```
birl (_, [x, mayis, Y | _], [o,s,m n, m, h, ---- ]
```

nisan — hatıra

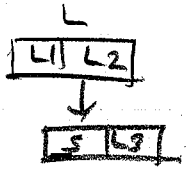
```
uye (X,L) :- birl (_, [x, _], L ).
```



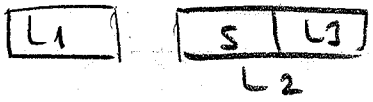
Alt Listeler :

```
alt ( s, L) :- birl (L1, L2, L), birl (s, L3, L2).
```

L: L1, L2 — L2: s, L3



OR] altlist (S, [1,2,3]), writelist(S), fail.



$L_2 = 1,2,3$

S	L3
ϕ	123
1	23
12	3
123	ϕ

$L_2 = 2,3$

S	L3
ϕ	2,3
2	3
23	ϕ

$L_2 = 3$

S	L2
ϕ	3
3	ϕ

$L_2 = \phi$

S	L2
ϕ	ϕ

S: ϕ 1, 12, 123, ϕ, 2, 23, ϕ, 3, ϕ

Yer değişmeler:

```
pre( [], [] ).
pre( [B|K], [X|Y] ):- pre(K,Y), yerd(B,X)
```

pre([siz, alman, misiniz], z)
↓ ↓ ↓
ben bilgisayar im

yerd(siz, ben)
yerd(alman, bilgisayar)
yerd(misiniz, im)

```
fib(0, 1, -)
fib(1, 1, 1)
fib(2, 2, 1)
fib(X, cevap, Eski) :- X>2, M=X-1, fib(M, Cevap1+Eski1),
                       Cevap=Cevap1+Eski1, Eski=Cevap1.
```

Tamsayı kare:

```
kareler ( [], [] ).
kareler ( [B|K], [S|YK] :- S=B*B , kareler ( K ,YK).
```

verilen 3lü komb. yazdırır.
123 124 234 134

```
cm (0, _, [] ).
cm ( N, [X|T], [X,Y] :- N>0, N1=N-1 , cm (N1 ,T, Y).
cm (N, [_|T] ,Y ) :- N<0 , cm (N,T,Y).
```

goal

```
cm (3, [1,2,3,4], X) write ("X =" ,X) . fail.
```

```
f1 ( X, [X|Y] , Y).
f1 ( X, [Y|Z], [Y|Z1]) :- f1 ( X,Z,Z1).
f2 ( [],[] ).
f2 (L, [X|P]) :- f1 (X,L,L1), f2 (L1 , P)
```

sil

yerdeğiş.

?f = ([k,a,r] ,S)

kar
kra
akr
ark
rka
rak

Son elemanı elde etme:

```
son([],Y)
son([X|K],Y):- son([K],Y)
```

```
sdz([],[])
sdz([B|K],[B|YK]):- sdz(K,YK).
```

sdz [2,3,6,2,3,2,6],Y

B=2
B=3
B=6
B=2
B=3
B=2
B=6

K= 3,6,2,3,2,6
K= 6 2 3 2 6