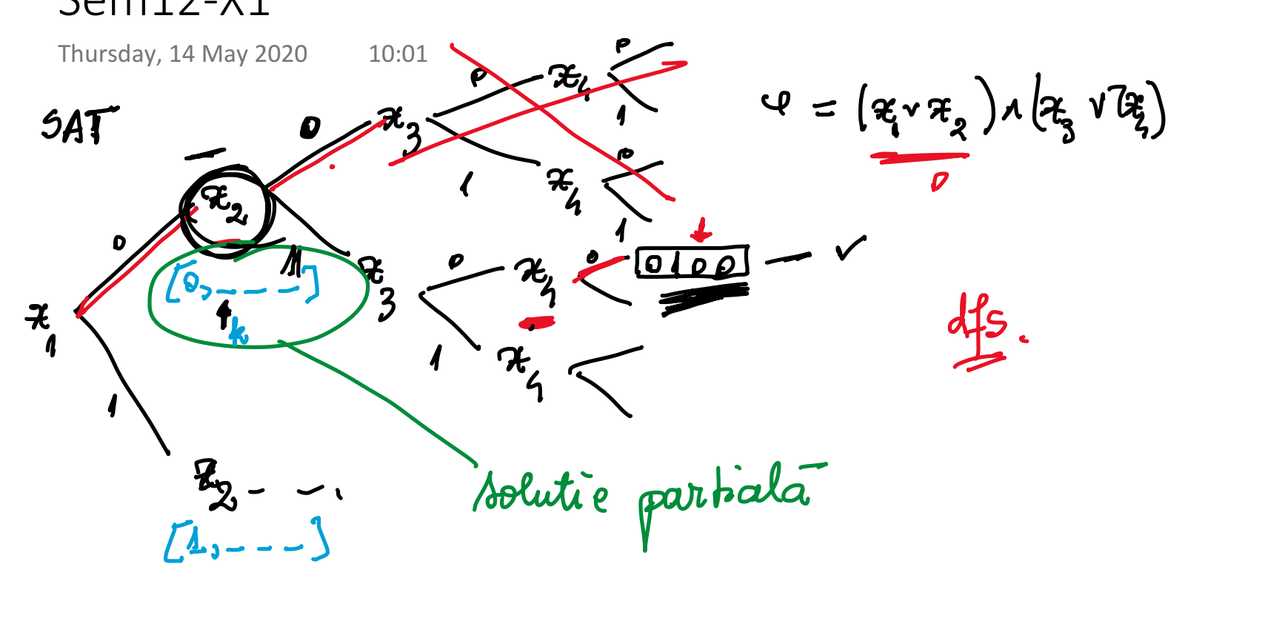
Sem12-X1

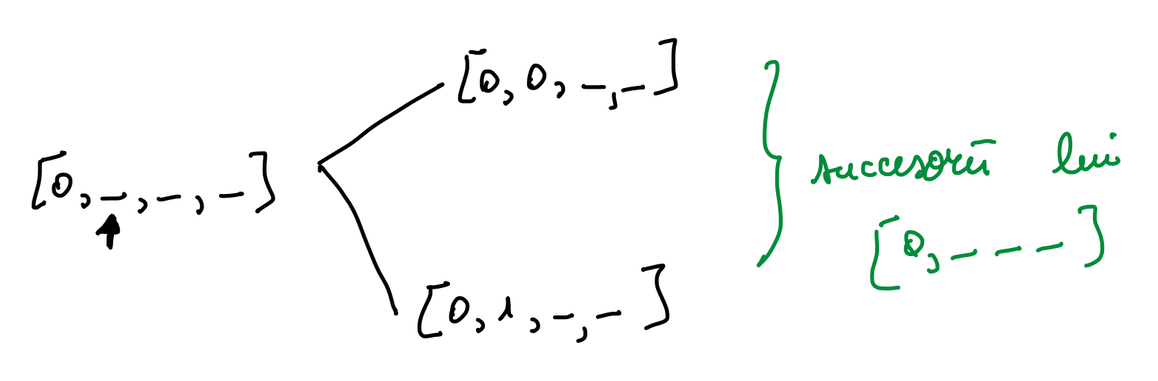
Thursday, 14 May 2020 10:01

SAT



$\varphi = (x_1 \vee x_2) \wedge (x_3 \vee \neg x_4)$

dfs.



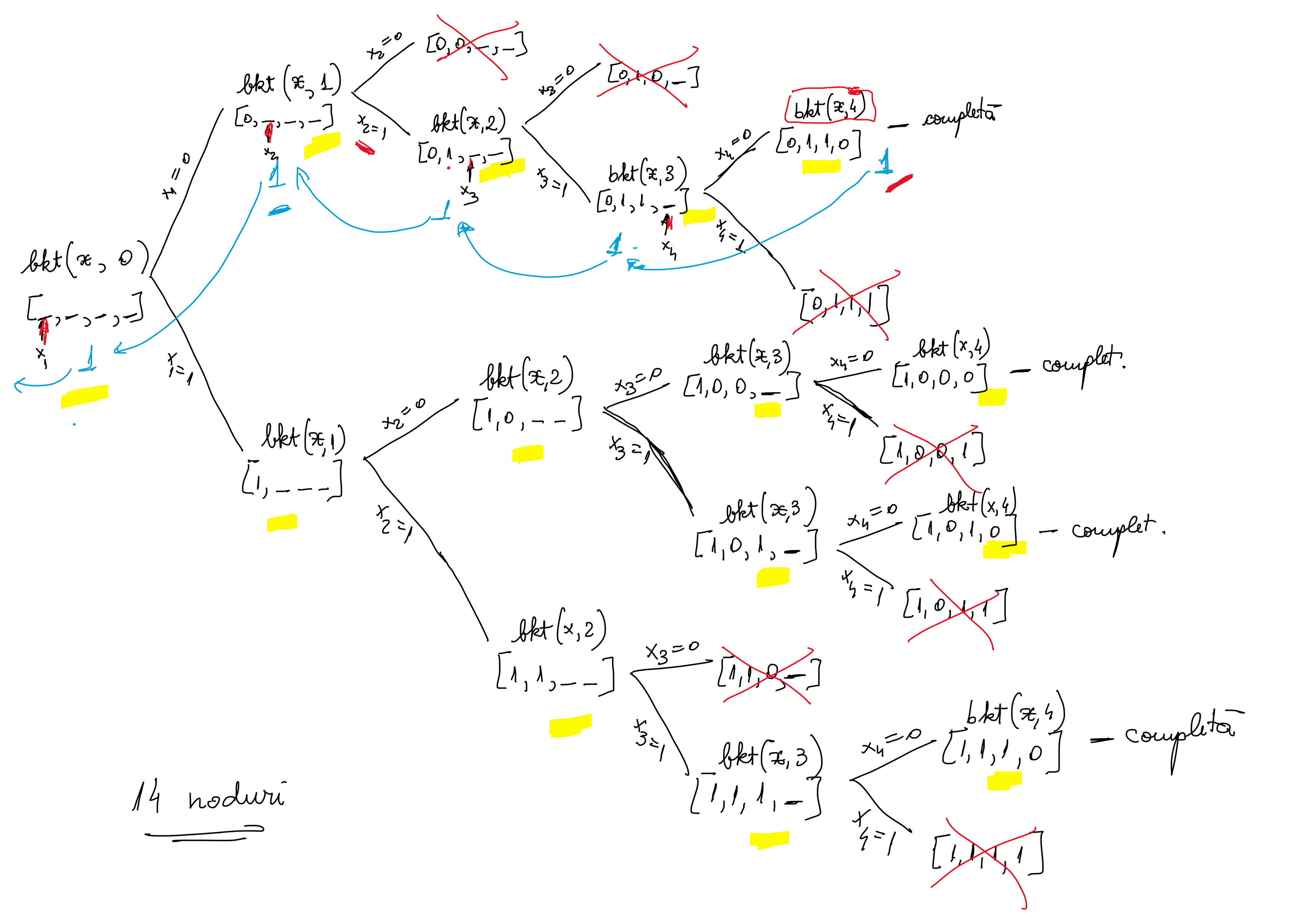

```
bkt(s)
{ if (s este completă)
  { print(s);
    return 1;
  }
  else { for (s' in succesor(s))
    { if (viabil(s'))
      { r = bkt(s');
        if (r==1) return 1;
      }
    }
    return 0;
  }
}
```

① $f = (x_1 \vee x_2) \wedge (\neg x_2 \vee x_3) \wedge \neg x_4$

soluția : $x = [0,1,0,1] \in \{0,1\}^4$

soluția parțială : $[0,1,|-,-]$ $k=2$

bkt(x[], k)



14 noduri

② 3-CNF-SAT

$\varphi = (x_1 \vee \neg x_2 \vee x_3) \wedge (\neg x_2 \vee x_4 \vee \neg x_5)$

$v[][] : \begin{pmatrix} 1 & -2 & 3 \\ -2 & 4 & -5 \end{pmatrix}$ $m = 2$

$x = [1,0,0,1,1]$

INPUT : m, $v[m][3]$

OUTPUT : $\exists ?\ x \in \{0,1\}^{n+1}$, $n = \max_{0 \le i < n,\ 0 \le j < 3} |v[i][j]|$ a.î.

$\prod_{i=0}^{m-1} \left(\sum_{j=0}^{2} \ell[i][j] \right) \neq 0$

$\ell[i][j] = \begin{cases} x[v[i][j]], & v[i][j] \ge 0 \\ 1 - x[|v[i][j]|], & v[i][j] < 0 \\ 1, & |v[i][j]| > k \end{cases}$

dacă avem valoare pt. $v[i][j]$, $|v[i][j]| \le k$

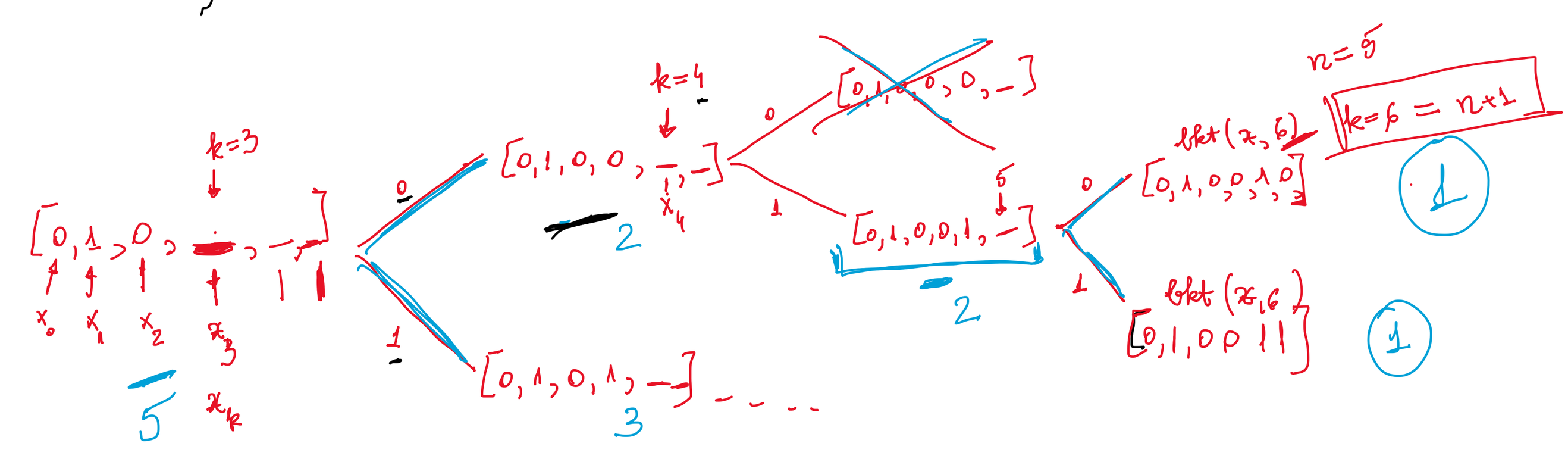
[0 ... n]

soluție : $x[] \in \{0,1\}^{n+1}$ $[0,1,...,n]$ ($x_0, x_1, \dots, x_n$)

soluția parțială : x[], k

$[0,1,0,-,-,-]$ $k=2$ (x_1, x_2)

```
viabil(x[], k, m, v[][])
{ for (i=0; i<m; i++)
  { s=0;
    for (j=0; j<3; j++)
    { if (v[i][j]>=0 && v[i][j]<=k)
        s += x[v[i][j]];
      if (v[i][j]<0 && abs(v[i][j])<=k)
        s += 1-x[abs(v[i][j])];
      if (abs(v[i][j])>k) s+=1;
    }
    if (s==0) return 0;
  }
  return 1;
}
```




```
         prop. curentă
bkt(x[], k, m, v[][], n)
{ if (k==n+1) {
    return 1;
  }
  else { for (a=0; a<=1; a++)
    { x[k]=a;
      if (viabil(x,k,m,v))
      { r = bkt(x,k+1);
        if (r==1) return 1;
      }
    }
    return 0;
  }
}
```

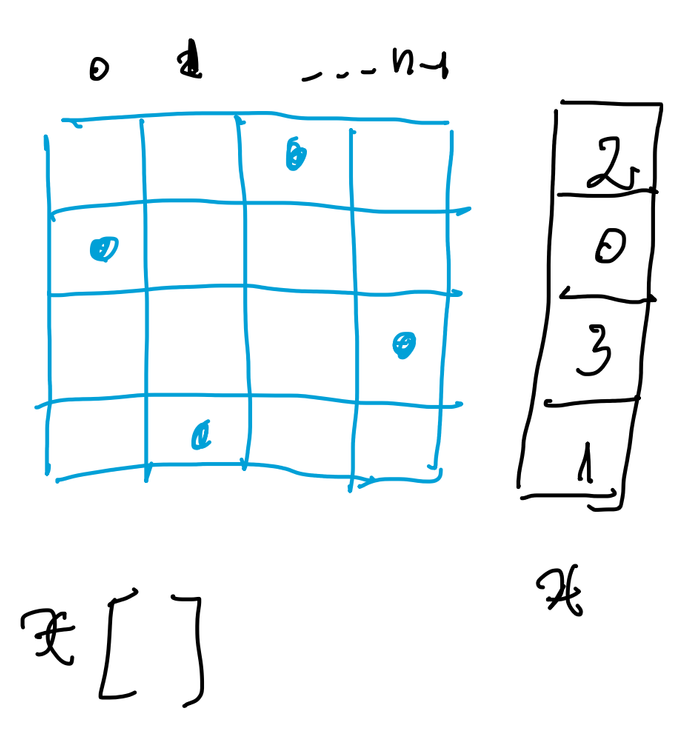
afișarea tuturor: print(x);

nr. soluțiilor: return 1; nr=0; r = nr. de soluții începând de la succesorii pt. a; nr += r; return nr;

```
sat3(m, v[][])
{ n=0;
  for (i=0; i<m; i++)
    for (j=0; j<3; j++)
      if (abs(v[i][j])>n) n=abs(v[i][j]);
  x=[0..n];
  r= bkt(x, 0, m, v, n);
  print(r);
}
```

③ problema celor n regine

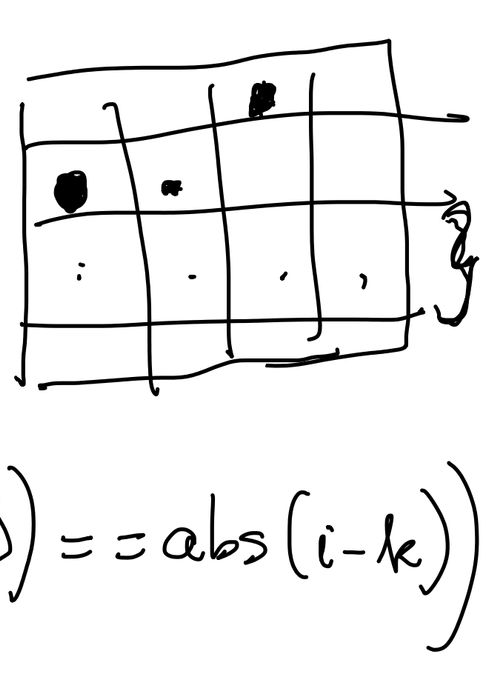
n = 4



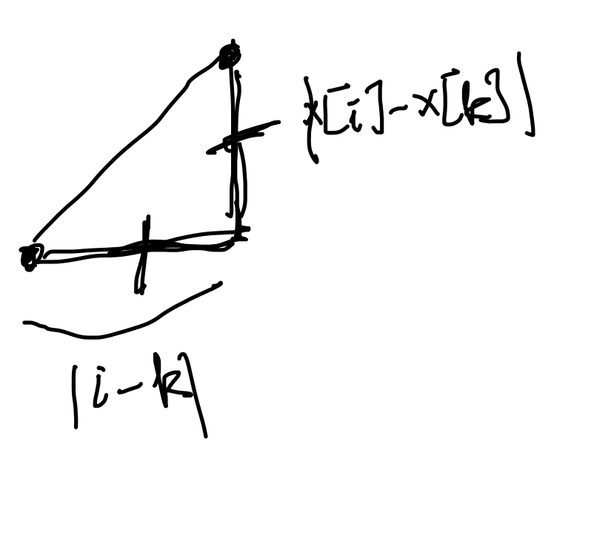
soluția : x[]

soluție parțială : x[], k

```
viabil(x[], k, a)
{ for (i=0; i<k; i++)
  { if (x[i]=x[k] || abs(x[i]-x[k])==abs(i-k))
      return 0;
  }
  return 1;
}
```



k=2

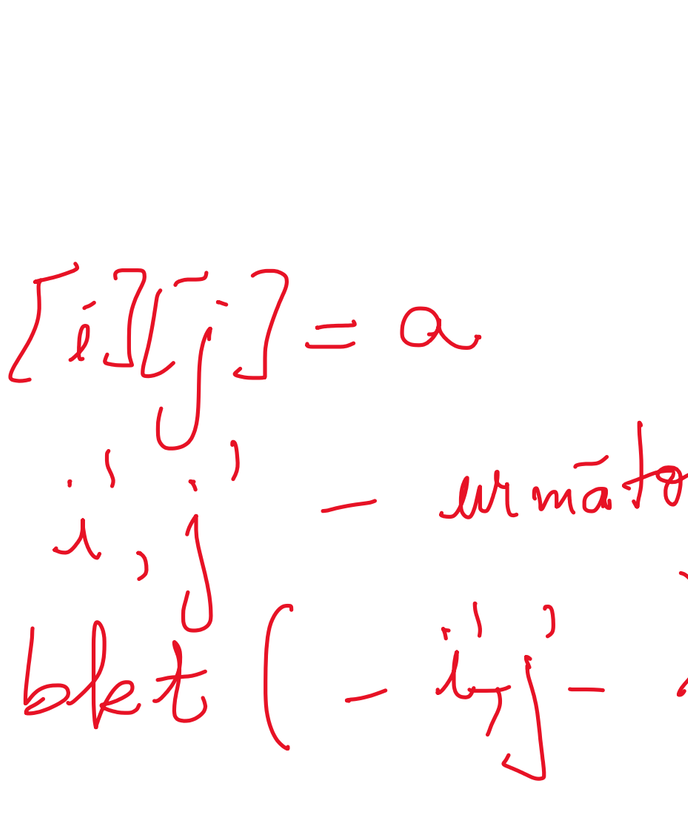



```
bkt(x[], k, n)
{ if (k==n) { print(x);
              return 1;
            }
  else { for (a=0; a<n; a++)
    { if (viabil(x[], k, a))
      { x[k]=a;
        r= bkt(x, k+1, n);
        if (r==1) return 1;
      }
    }
    return 0;
  }
}
```

[0 ... n-1] k=n

[0 ... k-1 k - - -]

Sudoku 1...4



x[i][j] = a

i', j' - următoarea poz. liberă

bkt(- i', j' -)

x[i][j] = -1