Sem12-B4

Thursday, 14 May 2020 08:00

```
sum ( n )
{
  s = 0;
  for( i=1 ; i<=n ; i++ )
  {  s = s+i;
  }
  return s;
}
```

$1+2+...+n = \frac{n(n+1)}{2}$

$s = 1+2+...+(i-1) = \frac{(i-1)\cdot i}{2}$ ← invariant

- $n=0 \Rightarrow$ nu execut for $\rightarrow s=0$
- pp. că $s=\frac{(i-1)\cdot i}{2}$ la intrarea în for.

i: - după o iteratie : ($s=s+i$)

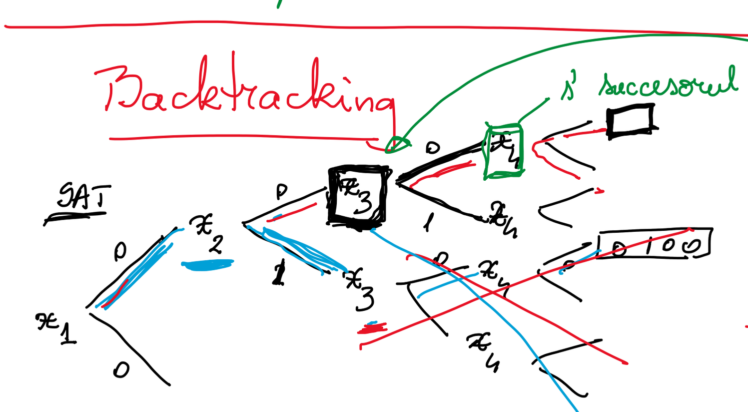
$s = \frac{(i-1)\cdot i}{2} + i = \frac{i(i+1)}{2}$ — la intrarea în for pt $i+1$

- ultima iteratie : $i=n$

după ultima iteratie $s=\frac{n(n+1)}{2}$

Backtracking

s' succesorul lui



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

0 0

0 soluția parțială $[0, 0, -, -]$

```
bkt( s )
{
  if ( s este completă )
  {      return 1;
  }
  else { for ( s' ∈ succesori ( s ) )
       {  if ( viabil( s' ) )   // poate duce la o solutie
          {  r = bkt ( s' );
             if( r == 1) return 1;
          }
       }
       return 0;
  }
}
```

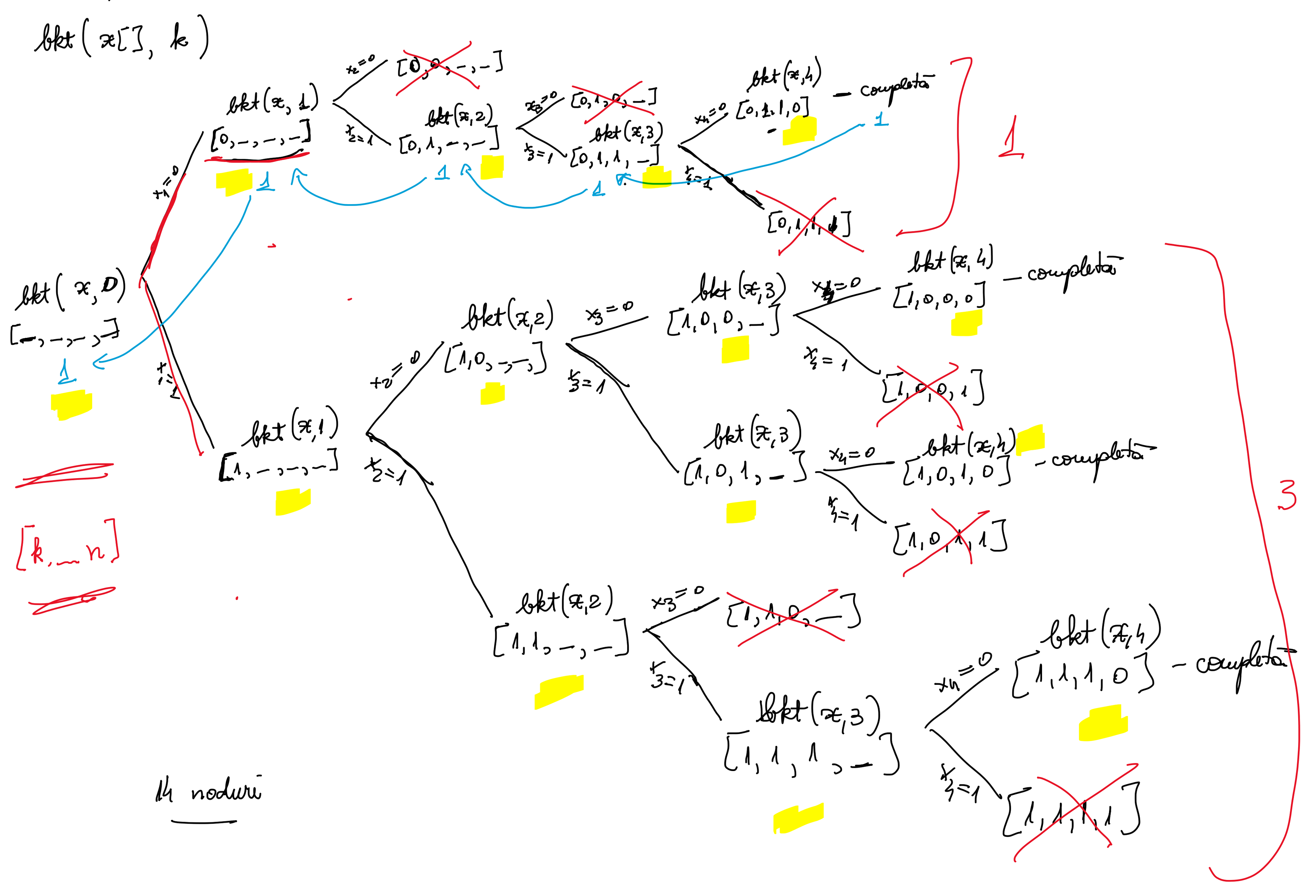
// nr++

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

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

soluție parțială : $[0, 1, -, -]$, $k=2$

bkt (x[ ], k)



14 noduri

② 3-CNF-SAT

INPUT : m, v[ ][ ]

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

$$\prod_{i=0}^{m-1} \left(\sum_{j=0}^{2} \ell[i][j] \right) \neq 0 \qquad \ell[i][j] = \begin{cases} x[v[i][j]-1], & v[i][j]>0 \\ 1 - x[abs(v[i][j])-1], & v[i][j]<0 \end{cases} \quad |v[i][j]| \le k$$

1 , altfel,

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

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

$k=2$

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

x_1 x_2

$0 .... n-1$

$$\begin{array}{ccccc} 0 & 1 & 2 & 3 & 4 \end{array}$$

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

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

nr. de soluții

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

– return 1

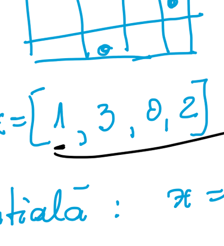
– s = 0

s += r

```
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-1];
   r=bkt ( x, 0, m, v, n );
     print(r);
}
```

③ problema celor n regine

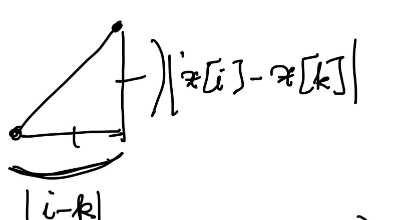
n = 4.



x[i] = linia reginei de pe coloana i

soluție : x = [1, 3, 0, 2]

soluția parțială : x = [1, 3, -, -] k = 2




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

bkt( x[ ], k, n )
{ if( k==n) return 1;
  else { for (a=0; a<n; a++)        — v[i][j] = a
         { x[k] = a;
           if( viabil ( x, k ))
           { r=bkt( x, k+1, n);
             if (r==1) return 1;
           }                         — v[i][j] = -1
         }                                eliberez
         return 0;
  }
}
```