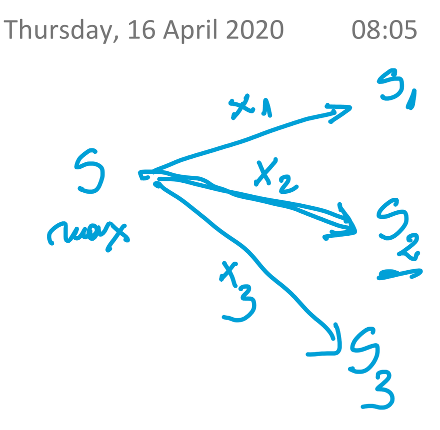
Sem9- B4

Thursday, 16 April 2020 08:05



$\max\left(x_1 + \underline{\quad}, x_2 + \underline{\quad}, \dots\right)$

$\downarrow$

$x_i + \underline{\quad}$

sol. optimă pt S_i

1a problema discretă a rucsacului

INPUT : n - nr. ob, $w[0 \dots n-1]$ - greutăți

$p[0 \dots n-1]$ - costuri

g - capacitatea rucsacului

OUTPUT : cost a.î. $a[0 \dots n-1] \in \{0,1\}^n$ a.î.

$$\sum_{i=0}^{n-1} w[i] \cdot a[i] \le g$$

$$cost = \sum_{i=0}^{n-1} p[i] \cdot a[i] \text{ - max}$$

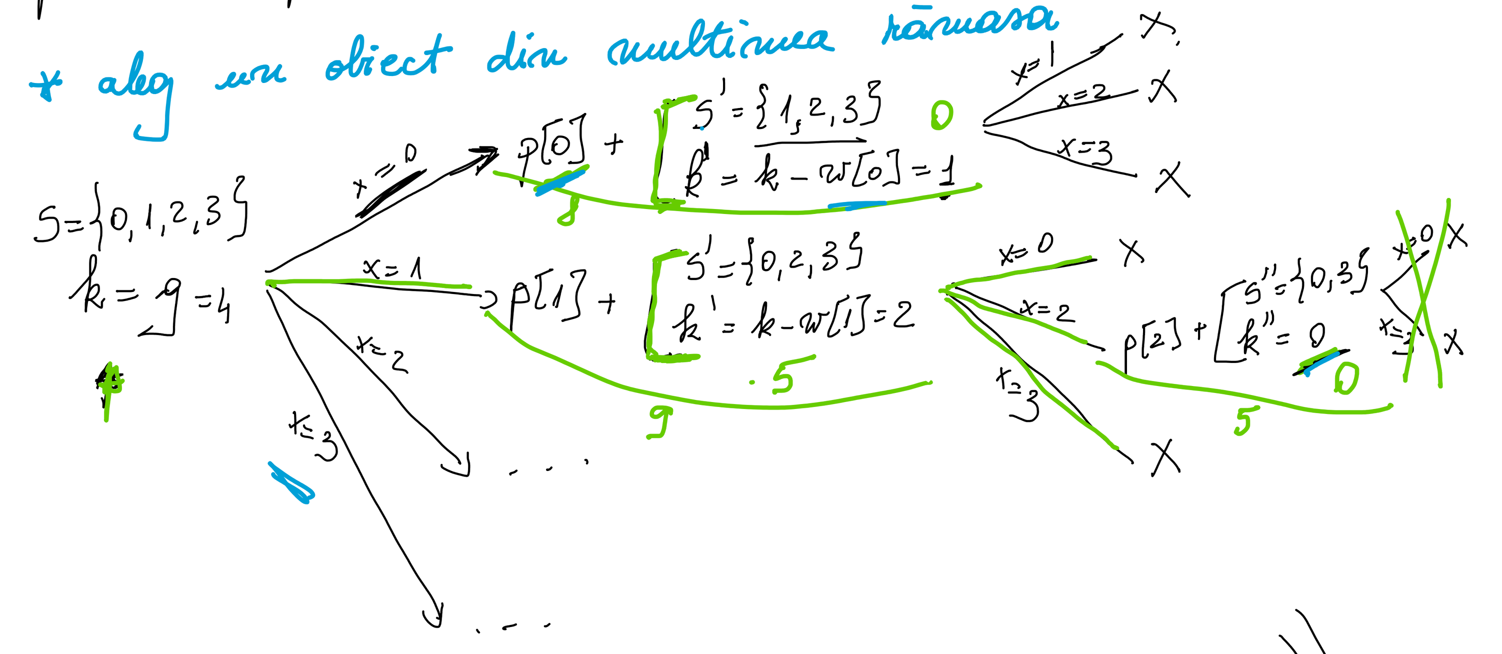
i	0	1	2	3
p	8	4	5	2
w	3	2	2	4
p/w	2,6	2	2,5	0,5

$g = 4$

max p/w $\to p[0] = 8$

optim $\to p[1] + p[2] = 9$

* alg un obiect din mulțimea rămasă



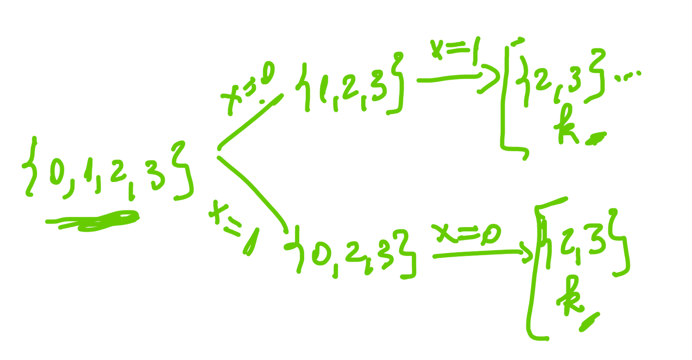
$$castig(S, k) = \max_{\substack{x \in S \\ w[x] \le k}} \left(p[x] + castig\left(S \setminus \{x\}, k - w[x]\right)\right)$$

{1,2,3}

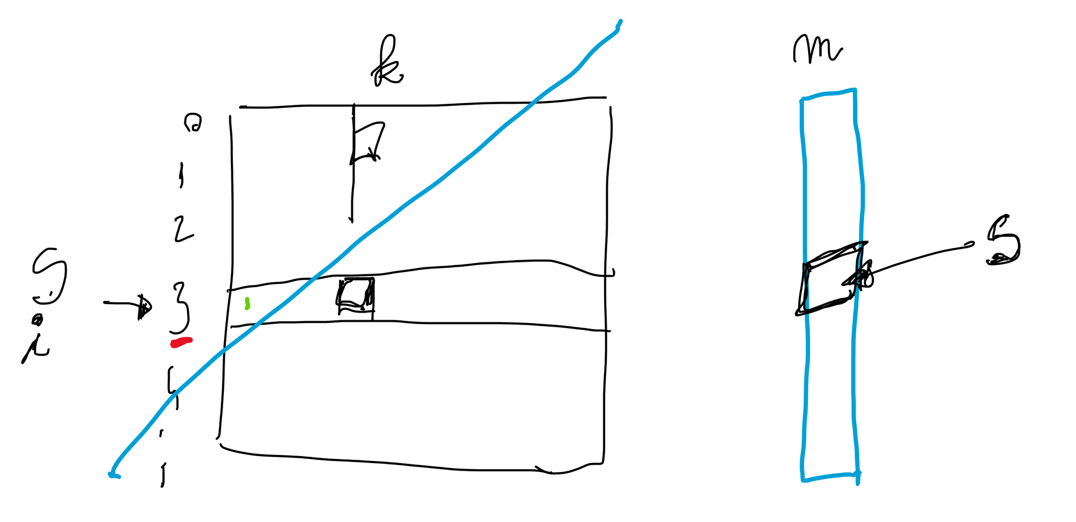
asociat cod! $\underline{0\ 1\ 1\ 1}$ → baza 10.

0 1 2 3

m[S] — castig(S, k, p, w) — inițial -1




```
castig ( S, k, p[], w[] )
{   if (k==0) return 0;
    max = 0;
    foreach x in S
    {   if ( w[x] <= k )
        {   v = p[x];
            if ( m[S] != -1 ) v = v + m[S];
            else v = v + castig( S \ singletonSet(x), k-w[x], p, w );
            if (v > max) max = v;
        }
    }
    m[S] = max;
    return max;
}
```



$S, k \longrightarrow k = g - \sum_{x \in A \setminus S} w[x]$

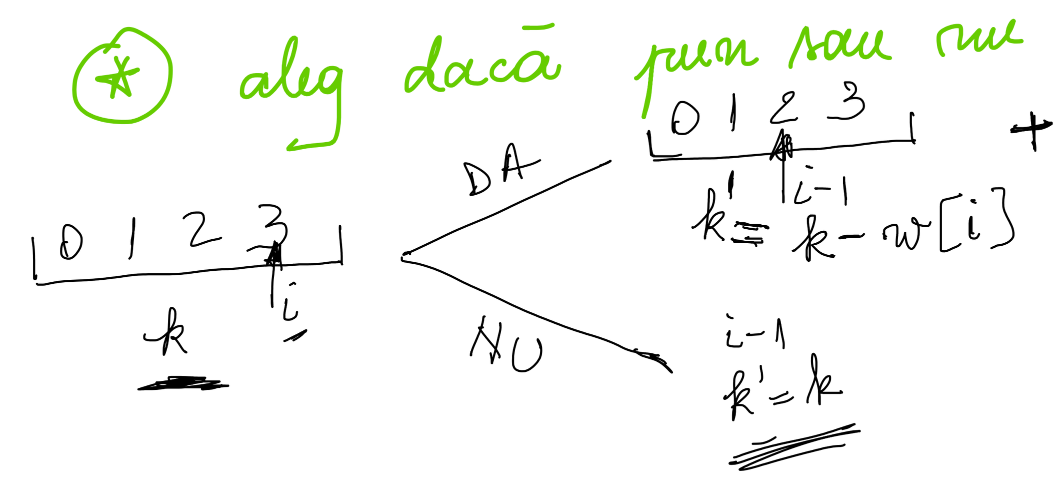
Probleme :

- codificarea lui S
- $|m| = O(2^n)$

{1,2,3} , k=4

{1,2,3} , k=3

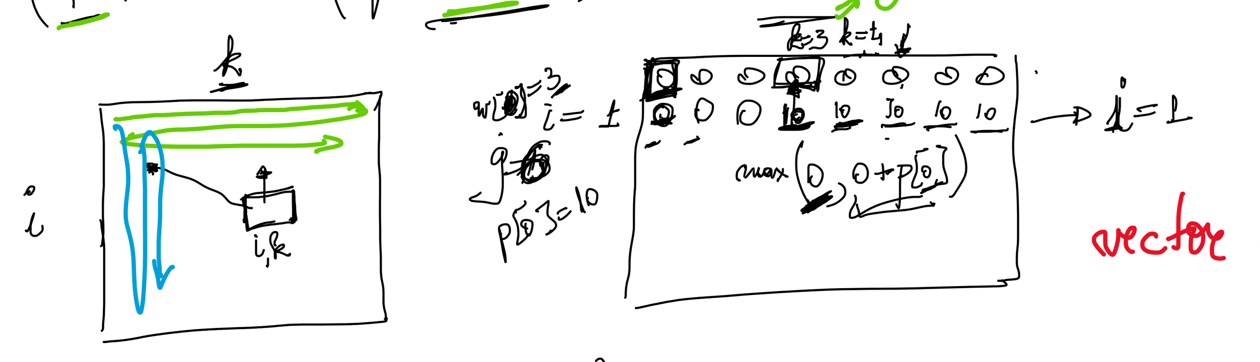
(*) alg dacă pun sau nu obiectul curent.



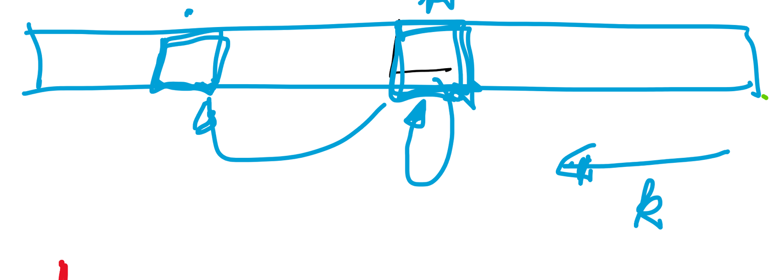
dp (i, k) = costul max pt. obiectele {0 ... i-1} în capacitatea k

$dp(i,k) = \max\left(dp(i-1, k), dp(i-1, k - w[i-1]) + p[i-1]\right)$

≥ 0



vector → linia i



d[i,k] — câștig dp(i, k)
— inițial -1 (necalculat)

```
castig (g, n, w[], p[])
{   for (j = 0; j <= g; j++)      // linia 0
        d[0,j] = 0;
    for (i = 1; i <= n; i++)
        for ( k = 0; k <= g; k++ )
        {   max = d[i-1, k];
            if (w[i-1] <= k && max < p[i-1] + d[i-1, k - w[i-1]])
                max = p[i-1] + d[i-1, k - w[i-1]];
            d[i,k] = max;
        }
    return d[n,g];
}
```

pt linia i → obiectul i-1

| d[j] = 0

// for(k=g ; k>=w[i-1] ; k--)

1b

INPUT : $S \subseteq \mathbb{Z}$

OUTPUT : DA dacă $\exists A \subseteq S$ a.î. $\sum_{x \in A} x = \sum_{y \in S \setminus A} y$

NU altfel

```
f ( S )
{   sum = 0;
    for ( x in S ) sum += x;
    if ( sum % 2 == 1 ) return 0;
    else // elementele din S în v[]
        part ( v, n, sum/2 ); // dacă există o submulțime cu suma sum/2
}
```

$\underbrace{sum/2}_{k}$

| 1 5 12 6 |, k = 12, i

DA true || NU false

$k = 0 \to \emptyset \to$ true

$k \ne 0,\ i = 0 \to$ false

$$d(i, k) = d(i-1, k - v[i-1]) \,||\, d(i-1, k)$$