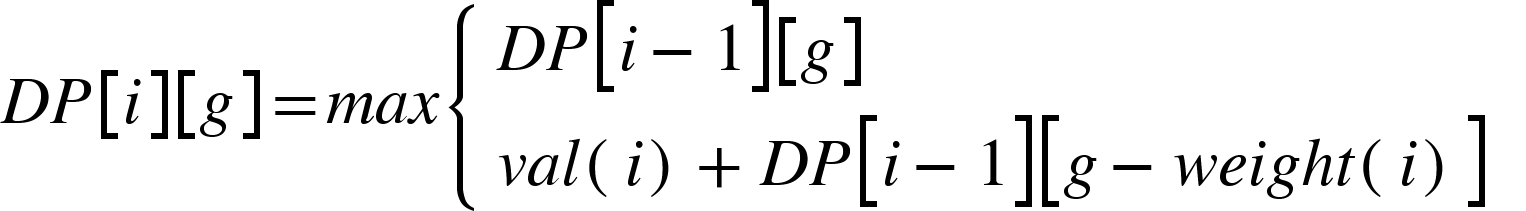
1/0 Knapsack problem:

* nu avem voie sa “taiem” obiectul

Programare dinamica:  
DP[i][g] - valoarea maxima care se poate se poate obtine alegand dintre primele i obiecte fara a depasi greutatea *g*

DP[i][0]=DP[0][g]=0

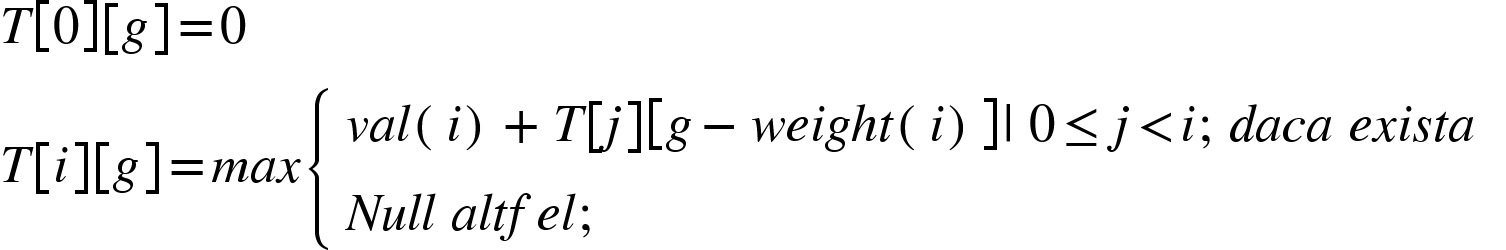


----------------------------------------

Cum ar arata o matrice T de forma:  
T[i][g] - valoarea maxima obtinuta astfel incat ultimul obiect adaugat in rucsac sa fie *i* si sa nu se depaseasca greutatea totala *g*

T[i][0]=-1

T[0][g]=0;



---------------------------------------------------------------

Input - citit din fisier:  
W=50 //capacitatea rucsacului

val: [60, 100, 120]

weight: [10, 20, 30]

Output:  
220; [0,1,1] -   
2,3

O(n\*W) - psudopolinomiala

Cand avem valori intregi pt greutatile obiectelor!

------------------------------------------------------------------------

2) Joc

Fie urmatorul joc de doua persoane.

Ambii jucatori au in fata lor un sir de numere S. Jucatorii muta succesiv astfel:  
Jucatorul curent alege unul dintre capetele sirului, “rupe” din sir numarul din acel capat iar valoarea sa o va adauga punctajului propriu. Jocul se termina cand situl S devine vid.

Jucatorul cu punctajul cel mai mare este castigator.

* implementarea jocului pe calculator:

1. Daca S este de lungime para, dati exemplu de o tactica simpla ce poate fi folosita de catre jucatorul 1 pt a obtine mereu cel putin scor egal.

* face suma de pe pozitii pare/impare.

T[i][j] - punctajul maxim ce poate fi obtinut de catre jucatorul aflat la mutare avand la dispozitie sirul S dar de la pozitia i la pozitia j

T[i][i]=S[i];

T[i][i+1]=max (S[i],S[i+1])

