1. 1/0 Knapsack Problem:

Avem o lista de obiecte reprezentate prin 2 numere: valoarea obiectului si greutatea sa.Se da W capacitatea unui rucsac. Trebuie sa aflam o submultime de obiecte astfel incat suma greutatilor obiectelor sa fie <= W, iar suma valorilor obiectelor sa fie maxima.

Input:

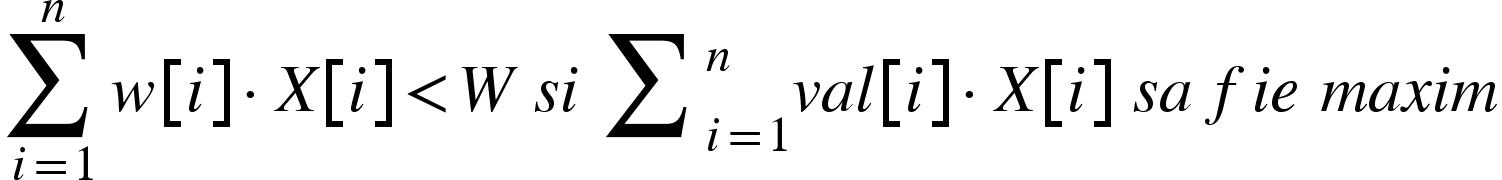
W - capacitatea rucsacului.

w[] - lista greutatatilor obiectelor; val[]-lista valorilor obiectelor; n- numarul de obiecte.

output: S - profitul maxim obtinut incarcand obiectele in rucsac

X[] - cu proprietatea ca X[i]=1 daca obiectul ‘i’ este incarcat in rucsac, respectiv 0 altfel.

Formal spus trenuie gasit X astfel incat:

 (erata: <=W)

Ex:

W=50

w={10,20,30}

val={60,100,120}

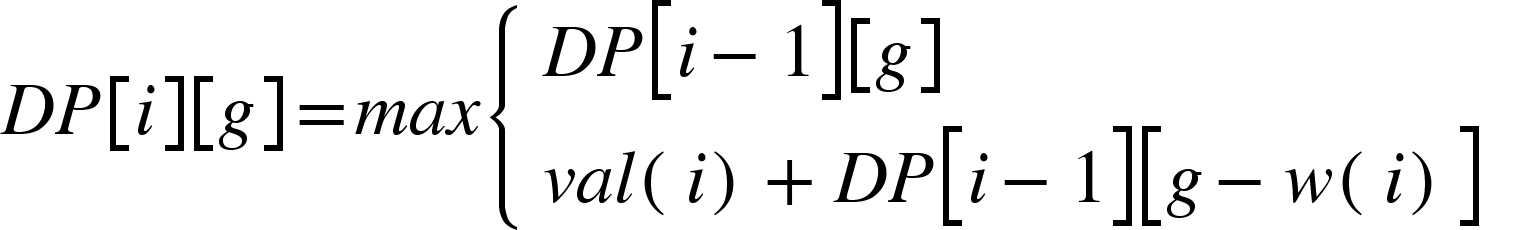
n=3 - numarul de obiecte

Output: 220, [0,1,1]

Rezolvare:  
DP[i][g]=profitul maxim obtinut alegand dintre primele ‘i’ obiecte fara a depasi greutatea ‘g’

DP[n][W] - raspunsul pentru problema initiala

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



T[i][g]=-1 - daca DP[i][g] vine de pe prima linie

T[i][g]=g-w(i) daca DPi][g] vine de pe a doua linie;

Problema 2:  
2 jucatori au in fata un sir de numere S.

La fiecare mutare jucatorul curent poate alege un numar dintre cele din capetele sirului, il “rupe” din sir, si numarul este adunat la punctajul jucatorului Jocul se termina cand sirul devine nul.

Trebuie sa gasim mutarile optime pt primul jucator.

T[i][j] - punctajul maxim ce poate fi obtinut de catre jucatorul curent avand in fata sirul S[i...j]

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

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

T[i][i+len]=? pt len>1

