Problema – **sumk**

Propunător: prof.Constantin Gălățan
Colegiul Naţional "Liviu Rebreanu", Bistrița

Soluţia 1 (50 puncte) prof. *Constantin Gălăţan*
Complexitate timp : O(N^4)
*Complexitate spatiu : O(N * N)*
Programare dinamică: se menţine un tablou `a` cu semnificaţia: `a[i][j][k]` – numărul de posibilităţi de a se ajunge în stagiul `i`, pe nivelul `j`, cu `k` puncte obţinute în stagiul `i`. Răspunsul se preia din `a[N][N][K]`.

Soluţia 2 (55 puncte) prof. *Marius Nicoli*
Complexitate timp : O(N^3)
*Complexitate spaţiu : O(N * N)*

Calculăm o matrice D [i][j] = numărul de posibilităţi de a finaliza stagiul i pe nivelul j.
D[i][j] = D[i-1][j] * B[1] [K]+ D[i-1][j-1]*B[2][K] + … D[i-1][1]*B[j][K].
Am notat B[i][j] = numărul de posibilităţi de a plasa j bile identice în i cutii, şi putem să avem cutii goale, mai puţin ultima cutie.
B[i][j] = C[i][j-1] (C[i][j] = numărul de posibilităţi de a plasa j bile identice în i cutii, şi orice cutie poate fi goală, considerând că fixăm o bilă în ultima cutie).
C[i][j] = C[i-1][j] + C[i][j-1], C[i][0] = 1, C[0][i] = 1. Matricea C se preprocesează în N^2.

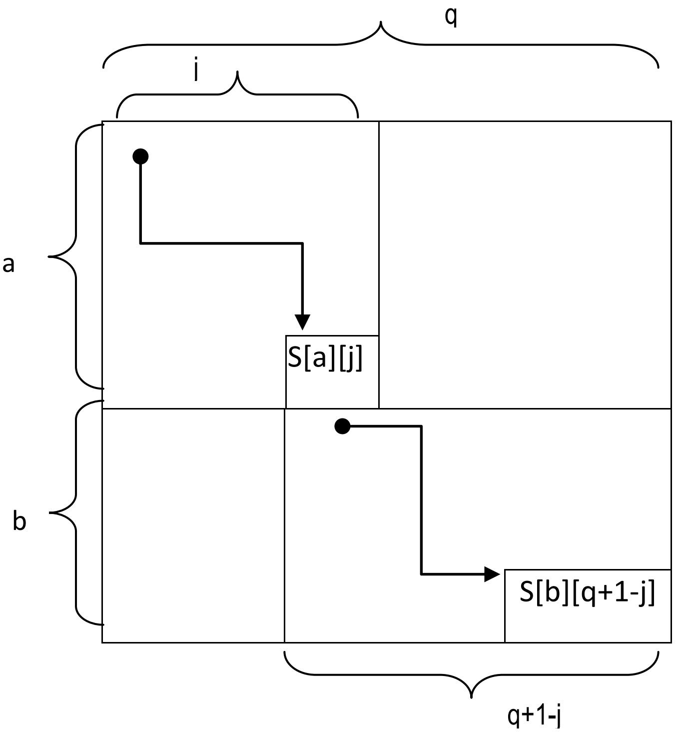
Soluția 3: (100 puncte) Prof. *Pit-Rada Ionel-Vasile*
Complexitate timp : O(N^2·log_2(N))
Definim s[p][q] = numarul modalitatilor de a termina un joc cu p stagii si avand pe q ca ultim nivel in stagiul p (in stagiul p ultimul termen nenul, prin care se va obtine suma k, este la pozitia q), unde 1<=p<=n si 1<=q<=n
Observam ca :

$$\mathrm{s[2p][q]} = \sum_{j=1}^{q} \mathrm{s[p][j]} * \mathrm{s[p][q+1-j]},$$

si avem in general

$$\mathrm{s[a+b][q]} = \sum_{j=1}^{q} \mathrm{s[a][j]} * \mathrm{s[b][q+1-j]}$$

intuitiv am putea reprezenta relaţia astfel:



Solutia problemei va fi numărul

$$\sum_{j=1}^{n} s[n][j]$$

Pentru a calcula linia s[n] vom calcula pe rand liniile s[1], s[2], s[4],s[8],…,s[2[log2n]] si vom utiliza doar acele linii ce corespund bitilor egali cu 1 din reprezentarea binara a lui n. (asemanator cu calculul rapid al puterii an). Astfel putem obtine complexitatea O(n2log2(n))

Pentru a calcula linia s[1] vom face urmatoarele observatii:
Definim c[i][j] = numarul secventelor de lungime i cu suma egala cu j si avand ultimul termen (al i-lea) nenul, unde 1<=i<=n si 0<=j<=k.
Se observa ca :
c[i][0]=0, pentru 1<=i<=n,
c[i][1]=1, pentru 1<=i<=n,

$$c[i][j] = \sum_{p=1}^{j-1} c[i-1][p]$$

$$c[i][j-1] = \sum_{p=1}^{j-2} c[i-1][p]$$

deci
c[i][j] = c[i][j-1] + c[i-1][j-1], pentru 1<=i<=n si 2<=j<=k,
Vom avea
s[1][q]=c[q][k], pentru 1<=q<=n,

Soluția 5 (100 puncte) *prof. Constantin Gălăţan*
*Complexitate timp : O(N * K)*
*Complexitate spaţiu : O(N*N)*
Să urmărim câteva variante posibile de terminare cu succes a jocului, pentru N = 3, K = 2:

Stagiul 1	1	1			1	1			1	0	1		0	0	2
Stagiul 2		2		...		2		...			2	...			2
Stagiul 3		1	1			0	2				2				2

Punctajul total va fi în final: N * K. Facem următoarea observație: vom asocia fiecărui punct obținut valoarea 1 și fiecărei deplasări spre dreapta (trecere la nivelul următor în același stagiu) valoarea 0. Pentru matricile din exemplu se vor obține șirurile:

10111101, 10111011, 10011111, 000111111

Avem deci N * K valori 1 și N – 1 valori 0 în fiecare șir. Se poate stabili deci o corespondență biunivocă între fiecare matrice soluție și șirul său asociat generat în acest mod. Se observă că numărul de șiruri este egal cu numărul combinărilor de N * K + N – 1 luate câte N – 1. Folosind inversul modular, această valoare se poate calcula în O(N * K).