



Problema 1 – cărți

Autor: Cosmin Tutunaru

Descrierea soluției (Cosmin Tutunaru)

Fie următoarele observații:

- 1) Întotdeauna are sens să plasăm cărțile la distanțe egale, astfel încât distanța maximă dintre oricare două cărți consecutive să fie minimă
- 2) Dacă plasăm k cărți pe raft se formează $k+1$ distanțe disponibile
- 3) Dacă vrem să verificăm dacă mai încap vreo carte pe raft are sens să o verificăm doar pe cea de grosime minimă
- 4) Fie k numărul cărților așezate pe raft, $1k$ suma lățimilor celor k cărți \Rightarrow avem o soluție validă dacă $(L-1k) / (k+1) <$ grosime minimă a unei cărți neașezată pe raft

Soluție de 60 pct

Sortăm cărțile după lungime, iar apoi selectăm pe rând cartea de grosime minimă pe care nu vrem să o punem pe raft, o vom numi cartea x . În mod evident, toate cărțile de grosime mai mică decât această carte le vom așeza pe raft. Pe lângă acestea, este posibil să aibă sens să mai punem pe raft și alte cărți, evident de grosime mai mare decât cea a cărții x .

Pentru a selecta în mod optim aceste cărți, vom face o dinamică $d_p[i][j] = \text{true}$ dacă există i cărți de grosime mai mare decât cea a cărții x , a căror grosime totală este exact j (problema rucsacului). Având această dinamică calculată, putem să încercăm pe rând toate combinațiile posibile și să le validăm conform formulei de mai sus.

Pentru a putea implementa eficient această soluție, este indicat să selectăm cartea x în ordine descrescătoare, astfel nu trebuie să reconstruim tabloul d_p de la început.

Complexitate: $O(N*N*L)$

Soluție de 100 pct

O îmbunătățire pe care o putem aduce algoritmului de mai sus ar fi să înlocuim tabloul $d_p[i][j]$ cu tabloul $best[i] =$ numărul minim de cărți (dintre cele de grosime mai mare decât a cărții x) a căror grosime totală este exact i . Restul observațiilor de la soluția anterioară se aplică și la această soluție.

Complexitate: $O(N*L)$