Programare dinamică

* Soluția problemei de dimensiune mare este exprimată în funcție de soluțiile unor subprobleme de dimensiune mai mică (deja calculate și stocate în tabela dp)

Exemple

**Gardurile lui Gigel:** Soluția pentru gard de dimensiune n este exprimată în funcție de soluțiile pentru gard de dimensiune n-1, respectiv pentru gard de dimensiune n-4

**Parantezare optimă de matrici:** Soluția pentru înmulțirea matricilor de la indexul i la indexul j este exprimată în funcție de soluțiile pentru Mi...Mk, respectiv Mk+1...Mj

Subșiruri de sumă pară

v[1..n] de numere naturale strict pozitive

Câte **subșiruri**(submulțimi nevide) au suma numerelor **pară**?

Exemplu:

1 3 2 4 1

Subșiruri de sumă pară: 2, 4, 1 3, 1 1, 3 1, 2 4, 1 3 2, etc.

Raționament:

* Încercăm ca data trecută: dp[i] = soluția (numărul de subșiruri de sumă pară) pentru v[1..i]
* Rezultă că dp[0] = 0 (există 0 submulțimi nevide în șirul vid)
* Este suficient? Se poate exprima dp[i] doar în funcție de subprobleme deja calculate? Cum?
  + Cum determinăm dp[1]?
  + Dar dp[2]?
* Nu este suficient! Avem nevoie să știm și câte subșiruri de sumă impară au fost până acum!
* Pentru aceasta vom ține 2 vectori:  
  pare[i] = numărul de subșiruri de sumă pară pentru v[1..i]  
  impare[i] = numărul de subșiruri de sumă impară pentru v[1..i]

Să vedem împreună cum evoluează cei 2 vectori:

v 1 3 2 4 1

pare 0 0 1 3 7 15

impare 0 1 2 4 8 16

Recurențele pentru cei 2 vectori sunt:

pare[i] = pare[i-1]\*2 + 1, daca v[i] par

= pare[i-1] + impare[i-1], daca v[i] impar

impare[i] = ...

Expresie booleană

expr = string de termeni și operatori booleeni (ex: „T&F^F^T”) (operatori posibili: & | ^)

n = lungimea stringului expr

În câte moduri se pot așeza paranteze în expr astfel încât rezultatul evaluării expresiei să fie **true?**

**Exemplu**

T&F^F^T

Moduri de parantezare: T&(F^(F^T)), T&((F^F)^T), (T&F)^(F^T), (T&(F^F))^T, ((T&F)^F)^T

Raționament:

* Încercăm ca la parantezarea de matrici:   
  dp[i][j] = soluția (numărul de moduri de a obține true pentru termenii Ti...Tj)  
  (ex: dp[3][7] = numărul de moduri de a obține true pentru parantezarea F^F^T)
* Rezultă că dp[i][i] = expr[i] == ‚T’ ? 1 : 0
* Este suficient? Se poate exprima dp[i][j] în funcție de intervale mai mici deja calculate? Cum?  
  dp[i][j] = f(dp[i][k], dp[k+2][j]) ?
* În ce fel contează operatorul de pe poziția intermediară?  
  T = ? & ? (T & T)  
  T = ? I ? (T | T, T | F, F | T)   
  T = ? ^ ? (T ^ F, F ^ T)
* Dacă sunt 3 moduri de a obține true în operandul stâng (L), respectiv 4 moduri de a obține true în operandul drept (R), câte moduri de a obține true în L&R există?