Le prob du sac à dos

// solution approximative. à voir plus tard.

→ maximiser 2 choses → souvent prob diffic

// val max pouvant être stocker dans le sac

⇒ ∃ 2 variants, avec/sans répétitions.

sans répétition :

$\alpha_{ij}$ = val max dans un sac de capacité $j$ avec les objets $1, 2 \dots i$

sol : $\alpha_{n,w}$

$\alpha_{0,j}$ = val max dans un sac de capacité $j$ avec des objets de l'ens. $\emptyset$.

= 0

$\alpha_{i,0}$ = val max dans un sac de capacité nulle

= 0

cas général : cas 1 : "poids objet $i > j$" $\underline{w_i > j}$ : $\alpha_{ij} = \alpha_{i-1,j}$

cas 2 : $w_i \leq j$

$$\alpha_{ij} = \max(\underbrace{v_i + \alpha_{i-1,\, j-w_i}}_{\text{on prend l'objet } i}, \underbrace{\alpha_{i-1,j}}_{\text{on ne prend pas}})$$

// NP-complet : si grand nbr marche pas.

exemple :

Init

tableau taille $(n+1) \times (w_{+1})$

Pour $i := 0$ à $n$ : $T[i][0] = 0$

Pour $j = 1$ à $w$ : $T[0][j] = 0$

Algo

Pour $i=1$ à $n$ :

  Pour $j=1$ à $w$ !

    si $\underbrace{P[i]}_{\text{poids objet}} > \underbrace{j}_{\text{capacité sac}}$ : $T[i][j] = T[i-1][j]$

    Sinon $T[i][j] = \max(V[i] + T[i-1][j-P[i]],\ T[i-1][j])$

| n° | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| w (P) | 2 | 8 | 3 | 4 |
| (v) | 5 | 22 | 8 | 5 |

nbr objets

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 |
| 2 | 0 | 0 | 5 | 5 | 5 | 5 | 5 | 5 | 22 | 22 (max(5 et 22+0)) | 27 | 27 | 27 |
| 3 | 0 | 0 | 5 | 8 | 8 | 13 | 13 | 13 | 22 | 22 | 27 | 30 | 30 |
| 4 | 0 | 0 | 5 | 8 | 8 | 13 | 13 | 13 | 22 | 22 | 27 | 30 | 30 |

// car pas répétitions.

// ici (v) écrit

// j à partir dbi ou réflécht

⤳ prog dynamique ⤳ possible optimiser sur 2 ligne de calcul, voir 1 si on fait ⚠

Avec économie de mémoire :

$[T[i][j] = \max(V[i] + T[i-1][j-P[i]],\ T[i-1][j])$ si $j \geq P[i]$

T = un tab de taille $w+1$

Pour $j=0$ à $w$ : $T[j]=0$

Pour $i=1$ à $n$ :

  Pour $j = w$ à $P[i]$ :

    $T[j] = \max(V[i] + T[j-P[i]],\ T[j])$

Complexité : $\Theta(n \cdot \underset{\geq}{w})$ ⚠ → non opti.

(avec répétition →) Pour j = w à P[i] ; ⇝ Pour j = P[i] à W

(avec répétition →)

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 5 | 5 | 10 | 10 | 15 | 15 | 22 | 22 | 27 | 27 | 32 |
| 2 | 0 | 0 | 5 | 8 | 10 | 13 | 16 | 18 | 22 | 24 | 27 | 30 | 32 |
| 3 | 0 | 0 | 5 | 8 | 10 | 13 | 16 | 18 | 22 | 24 | 27 | 30 | 32 |
| 4 | 0 | 0 | 5 | 8 | 10 | 13 | 16 | 18 | 22 | 24 | 27 | 30 | 32 |

} : TODO : Err

~ Un jeu ... ~

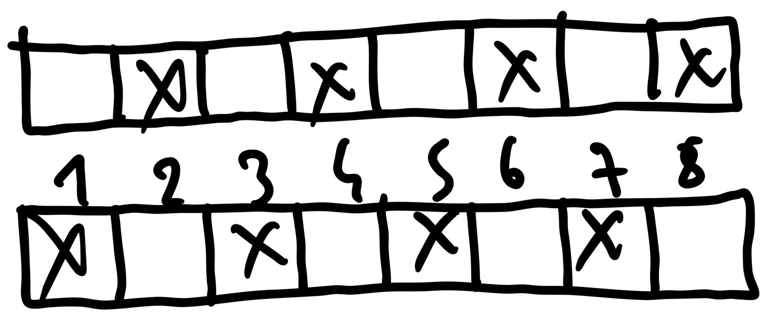
2 joueur : chacun choisit une carte à l'extrémité

gagnant celui somme des val prise est la plus grande

Stratégie gagnante :

- nbr impair cartes : 1er J pas sûr de gagner si [1 | 100 | 1] → J2 (V) ; [100 | 1 | 1] → J1 (V)
- nbr pair de cartes :

le J 1 ne peut jamais prendre



prendre soit que des cartes pair soit que impair