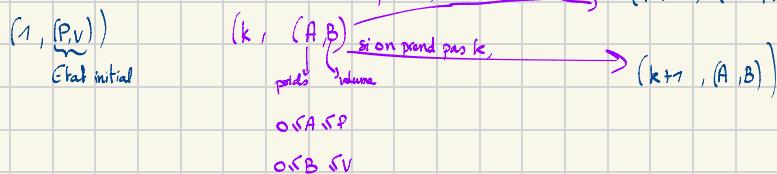


Pour le projet.

- programmer l'algo de l'algorithmique pour un sac à dos bi-dimensionnel (Poids et Volume) \Rightarrow reprendre les étapes du cours et convertir en code.
- programmer un générateur de jeu de données pour les tester sur l'algorithme.

- choisir une stratégie (longueur profondeur,...) pour notre algo
- Détaille en bête ce que l'algorithme doit faire et te donner à GPT.

Faire aussi un algo dynamique. \Rightarrow Convertir les 3 étapes du cours en code. Compléter E avec (A, B)



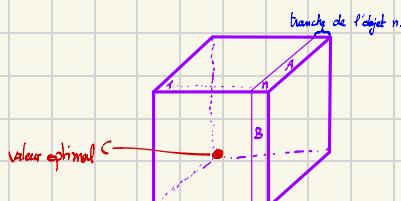
$$F_k(A, B) = \text{valeur maximale d'un chemin du sommet } (k, (A, B)) \text{ à un sommet } (n, X)$$

= valeur max d'un sac parmi les objets k à n qui pèse au plus A et a un volume au plus B.

$$F_n(A, B) = \begin{cases} W_n, & \text{si } P_n \leq A \text{ et } V_n \leq B \\ 0, & \text{sinon} \end{cases}$$

$$F_k(A, B) = \begin{cases} \max(V_k + F_{k+1}(A - P_k, B - V_k), F_{k+1}(A, B)) & \text{si } P_k \leq A \text{ et } V_k \leq B \\ F_{k+1}(A, B) & \text{sinon.} \end{cases}$$

Notre tableau a une troisième dimension.



Complexité de l'algo dans ce cas: $O(nPV)$

comparaison des perfos et sous de l'algo de l'arborescence et de l'algo dynamique.

- générateur abstrait d'intrana
- par instanciation des deux algos, s'assurer que le résultat est bon. Comparer les perfos.
Voir quel est le plus avantageux en fonction des instances testées.

Comparer par le temps d'exécution et les durées également entre les deux.