

Examen TPRO 2024 - 2025

Exercice 1 (4 pts)

Considérons la fonction suivante X , où $A[1..n]$ désigne un tableau et $[i..j]$ un intervalle $i \leq j \leq n$.

```
X(A, i, j): Si i < j alors
  X := X(A, i, j-1)
  Si A[j] < A[j-1] alors Permuter(A[j-1], A[j]) // Permuter A[j-1] et A[j] FinSi
FinSi X := A[j]
```

1. **Question** : Que fait cette fonction ? (1 pt)
 2. **Preuve** : Fournir une preuve de la correction de cette fonction en utilisant le système formel de Hoare. (2 pts)
 3. **Complexité** : Analyser et déterminer la complexité de cette fonction. (1 pt)
-

Exercice 2 (3 pts)

Transformer le **B-algorithme** suivant en un **D-algorithme** fonctionnellement équivalent.

B-algorithme :

```
1 : A; B; 2 : C; If T1 Goto 1; 3 : D; If T2 Goto 2; If T3 Goto 3; Stop
```

Exercice 3 (9 pts)

Partie 1 : Subset Sum Problem (SSP)

Considérez le problème SSP défini comme suit :

Entrée : Un ensemble $E = \{a_1, a_2, \dots, a_n\}$ d'entiers positifs et une somme S .

Question : Existe-t-il un sous-ensemble $E' \subseteq E$ tel que la somme des éléments de E' soit exactement égale à S ?

1. SSP est-il dans la classe **NP** ? Justifiez votre réponse. (1 pt)
2. Proposez une solution de type "force brute" ou exponentielle pour résoudre SSP. Fournissez un algorithme ou un pseudo-algorithme. (2 pts)
3. Peut-on appliquer la programmation dynamique au problème SSP ? Expliquez pourquoi. (1 pt)

4. Modifiez l'algorithme pour proposer une solution basée sur la programmation dynamique. (2 pts)

Partie 2 : Knapsack Problem

Considérez le problème **Knapsack** défini comme suit :

Entrée :

- Un ensemble A de n objets, chaque objet ayant un poids p et une valeur v .
- Un sac ayant une capacité maximale C .

Question : Trouver le sous-ensemble d'objets qui maximise la valeur totale sans dépasser la capacité C .

1. Knapsack est-il dans la classe **NP** ? Justifiez votre réponse. (1 pt)
 2. Montrez que SSP peut être réduit polynomialement au problème Knapsack. (2 pts)
-

Exercice 4 (2 pts)

1. Évaluer l'expression $S f g x$ où $S = \lambda x.\lambda y.\lambda z.(xz)(yz)$. (1 pt)
 2. Sachant que **True** : $T = \lambda x.\lambda y.x$, **False** : $F = \lambda x.\lambda y.y$, et **And** = $\lambda p.\lambda q.p q F$, évaluer $\text{And } T F$. (1 pt)
-

Exercice 5 (2 pts)

Écrivez un programme en **Prolog** qui détermine s'il existe un chemin entre deux nœuds dans un graphe orienté.

- Utilisez des faits pour définir les arêtes du graphe.
- Utilisez des règles pour déterminer s'il existe un chemin d'un nœud A à un nœud B . (2 pts)