Concours Commun INP (CCINP), France

Session 2021, filière TSI

Informatique

Durée : 3 heures

Ce que traite le sujet

- Partie I : optimisation du chargement
- Partie II : données liées aux livraisons

Sujet reproduit sans retouche à partir de la page suivante.

Résumé des parties, chapitres de cours liés et autres années :
prepahelp.com/annales/ccinp/tsi/2021/info

Prepa Help ne revendique aucun droit sur le sujet, propriété de l'organisateur du concours.

CONCOURS COMMUN INP

ÉPREUVE SPÉCIFIQUE - FILIÈRE TSI

INFORMATIQUE

Durée : 3 heures

N.B. : le candidat attachera la plus grande importance à la clarté, à la précision et à la concision de la rédaction. Si un candidat est amené à repérer ce qui peut lui sembler être une erreur d'énoncé, il le signalera sur sa copie et devra poursuivre sa composition en expliquant les raisons des initiatives qu'il a été amené à prendre.

RAPPEL DES CONSIGNES

- *Utiliser uniquement un stylo noir ou bleu foncé non effaçable pour la rédaction de votre composition ; d'autres couleurs, excepté le vert, peuvent être utilisées, mais exclusivement pour les schémas et la mise en évidence des résultats.*
- *Ne pas utiliser de correcteur.*
- *Écrire le mot FIN à la fin de votre composition.*

Les calculatrices sont interdites.

Le sujet est composé de deux parties indépendantes.

Le sujet comporte :

- le texte du sujet : page 1 à page 6 ;
- le Document Réponse : page 1 à page 11.

Le Document Réponse doit être rendu dans son intégralité.

A

Optimisation de rendement d'une entreprise de livraison

Une entreprise de livraison dispose de plusieurs locaux en France et chacun possède plusieurs camions de livraisons. Celle-ci souhaite optimiser le chargement de ses camions pour diminuer ses frais de fonctionnement.

Partie I - Optimisation du chargement

Chaque camion de l'entreprise peut charger une cargaison jusqu'à un poids maximal noté P_{max}. L'entreprise dispose de différentes informations provenant de ses clients :

- le poids de chaque produit p_i (chaque client propose un seul produit) ;
- la valeur v_i associée au transport de chaque produit : c'est-à-dire l'argent gagné par l'entreprise si elle réalise le transport de ce produit.

En considérant que l'entreprise dispose de n clients, l'entreprise cherche donc à trouver une liste d'indices notée I contenue dans $\{1, \ldots, n\}$ telle que :

$$\sum_{i \in I} p_i \leq \mathrm{P}_{max} \quad \text{(respect du poids maximal)} \tag{1}$$

et

$$\sum_{i \in I} v_i \text{ soit maximal} \quad \text{(optimation du profit pour l'entreprise).} \tag{2}$$

Dans toute la suite, les poids seront donnés en centaines de kilogrammes et les valeurs en centaines d'euros.

I.1 - Un exemple

Dans cette sous-partie, on suppose que $n = 4$ et que $\mathrm{P}_{max} = 8$ centaines de kilogrammes. On stocke alors les différentes informations dans trois listes :

- `Pr` est la liste des produits proposés par les clients numérotés de 1 à 4 : `Pr = [1,2,3,4]` ;
- `P` est la liste des poids associés : `P = [3,2,1,4]` ;
- `V` est la liste des valeurs associées : `V = [4,3,1,9]`.

Par exemple, le produit 2 a un poids de deux centaines de kilogrammes et une valeur de trois centaines d'euros.

Les **questions Q1**, **Q2** et **Q3** se traitent à l'aide de calculs simples, à faire à la main.

Q1. Expliquer pourquoi une cargaison constituée d'un, de deux ou de quatre produits ne répond pas au problème posé, c'est-à-dire ne maximise pas le profit fait par l'entreprise en respectant la condition donnée sur le poids maximal.

Q2. Donner toutes les cargaisons de trois produits respectant le poids maximal. On donnera à chaque fois le profit fait par l'entreprise.

Q3. Quelle est la cargaison maximisant le profit de l'entreprise ? Que vaut le profit dans ce cas ?

I.2 - Une méthode intuitive pour la résolution du problème

On garde les notations de la sous-partie précédente dans le cas général :

- `Pr` est la liste des produits (numérotés de 1 à n inclus) ;
- `P = [`p_1`,..., `p_n`]` est la liste des poids associés aux produits ;
- `V = [`v_1`, ..., `v_n`]` est la liste des valeurs associées aux produits.

Q4. Définir une fonction `ListeProduits` ayant pour argument un entier naturel non nul n et renvoyant la liste `Pr`.

Une méthode intuitive pour tenter d'optimiser le profit de l'entreprise est la suivante : on calcule les ratios $\frac{v_i}{p_i}$, puis on trie les objets par ordre décroissant suivant ces valeurs. Les produits sont alors classés par rentabilité : le premier produit devient le plus rentable " au poids " et ainsi de suite. On ajoute progressivement chaque produit dans la cargaison, dans cet ordre, sans dépasser la limite du poids maximal.

Q5. Définir une fonction `Ratio` ayant pour arguments deux listes `P`, `V` où `P` correspond à la liste des poids et `V` correspond à la liste des valeurs, renvoyant la liste des ratios $\frac{v_i}{p_i}$.

La fonction suivante est associée à une méthode de tri :

```
1  def Tri(L):
2  ''' L est une liste de nombres réels '''
3      for i in range(1,len(L)):
4          x=L[i]
5          j=i
6          while j>0 and x<L[j-1]:
7              L[j]=L[j-1]
8              j=j-1
9          L[j]=x
10     return L
```

Q6. On exécute `Tri(L)` avec `L=[3,5,2,1]`. Combien y a t-il d'itérations de la boucle for ? Donner la valeur de `L` à la fin de chaque itération de la boucle for.

Q7. Ce tri fonctionnerait-t-il pour une chaîne de caractères dont chaque caractère est un entier ? Justifier.

Q8. Quelle est la méthode de tri utilisée dans la fonction `Tri` ? Donner sa complexité (en nombre de comparaisons) dans le pire des cas et dans le meilleur des cas. *On justifiera soigneusement les complexités données.*

Q9. Définir une fonction `Inverse` ayant pour argument une liste de nombres réels `L` et renvoyant l'inverse de celle-ci. Par exemple, l'inverse de $[1, 5, 3, 4]$ est $[4, 3, 5, 1]$.

L'utilisation de `L[::-1]` n'est pas autorisée.

Q10. On souhaite trier une liste de poids `P` et une liste de valeurs `V` associées à une liste de produits en suivant l'ordre décroissant de la liste des ratios $\frac{v_i}{p_i}$. Justifier que les fonctions `Ratio`, `Tri` et `Inverse` ne permettent pas de répondre simplement au problème posé.

Q11. Écrire, à l'aide des fonctions `Ratio` et `Inverse`, une fonction `Tri2` ayant pour arguments une liste de poids `P` et une liste de valeurs `V` associées à une liste de produits. Cette fonction renverra les listes de poids et de valeurs triées par ordre décroissant de la liste des ratios.

Q12. Compléter la définition de la fonction `Vmax` ayant pour arguments les listes de poids `P` et de valeurs `V` et le poids maximal P_{max} du chargement et renvoyant la valeur maximale du profit de l'entreprise en suivant la méthode proposée.

Q13. On souhaite appliquer cette méthode en utilisant les listes de poids et de valeurs de la sous-partie **I.1**. Donner la liste des ratios, les listes de poids et de valeurs obtenues à l'aide de la fonction `Tri2` ainsi que le profit obtenu. Commenter le résultat.

I.3 - Une méthode récursive

Nous gardons les notations du cas général de la sous-partie **I.2** : `Pr`, `P` et `V`. On considère pour simplifier que les poids des produits sont des entiers, ainsi que P_{max}.

Nous introduisons une méthode récursive pour résoudre le problème d'optimisation :

- pour chacun des produits, deux choix sont possibles : il fait partie de la cargaison ou non ;
- la récursivité s'effectuera sur la liste des indices de `Pr` : le premier appel de la fonction se fera en utilisant l'indice n, puis l'indice $n-1$ et ainsi de suite jusqu'à l'indice 0 (correspondant au cas où il n'y a plus de produits) ;
- pour $i \in \{0, 1, \ldots, n\}$ et $\omega \in \{0, 1, \ldots, P_{max}\}$, on note $S(i, \omega)$ la valeur maximale cumulée des produits que l'on peut placer dans un camion d'une capacité maximale (en poids) de ω avec la liste constituée des i premiers produits de `Pr`.

On pose alors la relation de récursivité suivante :

$$S(i,\omega) = \begin{cases} 0 & \text{si } i = 0 \\ S(i-1,\omega) & \text{si } i > 0 \text{ et } p_i > \omega \\ \max(S(i-1,\omega), v_i + S(i-1,\omega-p_i)) & \text{si } i > 0 \text{ et } p_i \leq \omega. \end{cases} \quad (3)$$

Q14. Justifier les relations précédentes dans les trois cas.

Q15. Justifier la terminaison de l'algorithme associé à la relation de récursivité précédente, sachant que la première valeur donnée pour i sera n et la première valeur pour ω sera `Pmax`.

Q16. Définir une fonction `Max` ayant pour arguments deux réels et renvoyant le maximum parmi ces deux valeurs. Il est interdit d'utiliser la fonction `max` prédéfinie dans Python.

Q17. En vous basant sur la relation (3), compléter la définition de la fonction récursive `recur` ayant pour arguments les listes de poids et de valeurs `P` et `V`, un indice i, un poids ω, et renvoyant $S(i, \omega)$.

Q18. Donner une série d'instructions utilisant la fonction `recur` et permettant de déterminer le profit de la sous-partie **I.1**.

I.4 - Amélioration de la méthode récursive

On souhaite améliorer la méthode récursive de la sous-partie **I.3**. Nous allons procéder en mémorisant des calculs déjà effectués. Voici le principe : nous allons stocker les valeurs $S(i, \omega)$ dans un tableau `Memoire`, de taille $(n+1) \times (P_{max}+1)$, initialisé au départ avec des éléments tous égaux à -1.

Si la valeur de $S(i,\omega)$ a déjà été calculée, l'élément d'indice (i,ω) de ce tableau `Memoire` ne sera plus égal à -1 : on renverra donc directement la valeur. Sinon, on la calculera en suivant le principe de la sous-partie **I.3** et on la stockera dans le tableau avant de la renvoyer.

Nous faisons le choix de représenter les tableaux comme des listes de listes.

Q19. Donner l'instruction permettant de créer le tableau `Memoire` initialisé avec des coefficients égaux à -1, en supposant `Pmax` et `n` connus.

Q20. Compléter la fonction `recur2` en suivant le principe expliqué et permettant d'améliorer la fonction `recur`. La variable `Memoire` sera utilisée comme une variable globale.

Avec les données suivantes :

- `P = [5,3,3,3]` ;
- `V = [4,3,1,1]` ;
- $P_{max} = 8$;
- `Memoire` un tableau de taille 5×9 ;

et en exécutant `recur2(P,V,len(P),Pmax,Memoire)`, on obtient alors la valeur 7.

Partie II - Données liées aux livraisons conservées par l'entreprise

À chaque livraison, l'entreprise stocke des données relatives à celle-ci. L'entreprise dispose de 20 locaux, numérotés de 1 à 20, disposant chacun d'un certain nombre de camions. Pour faciliter ses livraisons, l'entreprise découpe la France en 30 zones et associe à chaque local, trois zones possibles de livraisons.

Ces données sont enregistrées dans une base de données contenant trois tables :

La table **livraison** constituée des champs suivants :
- *date* : date de la livraison au format "jj-mm-aaaa" (chaine de caractères) ;
- *heure* : heure de la livraison au format : "hh-mm-ss" (chaine de caractères) ;
- *id_client* : identifiant du client recevant la livraison (entier) ;
- *id_local* : identifiant du local de l'entreprise (entier compris entre 1 et 20).

La table **client** constituée des champs suivants :
- *id* : identifiant du client (entier) ;
- *zone* : entier compris entre 1 et 30.

La table **local** constituée des champs suivants :
- *id* : identifiant du local (entier compris entre 1 et 20) ;
- *zone1* : entier ;
- *zone2* : entier ;
- *zone3* : entier.

Q21. Donner une clé primaire pour la table **livraison**.

Q22. Écrire une requête SQL permettant d'obtenir les identifiants des clients livrés le 10 janvier 2021.

Q23. Écrire une requête SQL permettant de récupérer les dates et les heures de toutes les livraisons ayant eu lieu dans la zone 5 le 2 mars 2021.

Q24. Écrire une requête SQL permettant de compter le nombre de livraisons effectuées le 3 février 2021 par des camions dont les locaux ne livrent que dans des zones possibles inférieures ou égales à dix.

Chaque identifiant de client est stocké par l'entreprise en codage binaire (avec 8 bits). Par exemple, 00010111 est associé au client dont l'identifiant est le numéro 23.

Q25. Donner le codage associé au client 39.

Afin de retrouver l'identifiant de chaque client à l'aide de son code binaire, la fonction suivante est proposée :

```
1  def Identifiant(Bin):
2      '''Bin est est une chaine de caractères
3       constituée de 0 et 1'''
4      S=0
5      for i in range(len(Bin)):
6          S=S+Bin[i]*2**(len(Bin)-i)
7      return S
```

Q26. Trouver les deux erreurs dans le code de la fonction précédente.

On suppose maintenant la fonction précédente corrigée. La ligne 6 pose un problème de complexité : à chaque itération de boucle, la puissance de 2 est recalculée entièrement.

Q27. Écrire une fonction `Identifiant2`, qui donne le même résultat que la fonction `Identifiant` avec une meilleure complexité. Le nombre de multiplications devra être linéaire suivant la longueur de `Bin`.

Q28. Si l'entreprise souhaite aussi stocker la zone de chaque client en codage binaire, donner le nombre de bits minimal nécessaire.

FIN

IMPRIMERIE NATIONALE – 21 1177 – D'après documents fournis

CONCOURS COMMUN INP

Numéro d'inscription

Numéro de table

Né(e) le

Nom :

Prénom :

Emplacement QR Code

Filière : TSI

Session : 2021

Épreuve de : INFORMATIQUE

Consignes

- *Remplir soigneusement l'en-tête de chaque feuille avant de commencer à composer*
- *Rédiger avec un stylo non effaçable bleu ou noir*
- *Ne rien écrire dans les marges (gauche et droite)*
- *Numéroter chaque page (cadre en bas à droite)*
- *Placer les feuilles A3 ouvertes, dans le même sens et dans l'ordre*

TSI5IN

DOCUMENT RÉPONSE

Ce Document Réponse doit être rendu dans son intégralité.

Optimisation de rendement d'une entreprise de livraison

Q1

. . . . . .

. . . . . .

. . . . . .

. . . . . .

. . . . . .

. . . . . .

. . . . . .

NE RIEN ÉCRIRE DANS CE CADRE

Q2

.....

.....

.....

.....

.....

.....

Q3

Cargaison maximisant le profit : .....

Profit maximal : .....

Q4

```
def ListeProduits(n) :
```

Q5

```
def Ratio(P,V) :
```

Q6

Nombre d'itérations de la boucle for : .....

Valeur de L après chaque itération :

.....

.....

.....

.....

Q7

.....

.....

Q8

Méthode de tri utilisée : .....

Complexité dans le pire des cas : .....

Complexité dans le meilleur des cas : .....

Justifications : .....

.....

.....

.....

.....

.....

.....

.....

.....

Q9

```
def Inverse(L) :
```

Q10

.....

.....

.....

.....

CONCOURS COMMUN INP

Numéro d'inscription

Numéro de table

Né(e) le

Nom :

Prénom :

Emplacement QR Code

Filière : TSI

Session : 2021

Épreuve de : INFORMATIQUE

Consignes

- *Remplir soigneusement l'en-tête de chaque feuille avant de commencer à composer*
- *Rédiger avec un stylo non effaçable bleu ou noir*
- *Ne rien écrire dans les marges (gauche et droite)*
- *Numéroter chaque page (cadre en bas à droite)*
- *Placer les feuilles A3 ouvertes, dans le même sens et dans l'ordre*

Q11

```
def Tri2(P,V) :
```

NE RIEN ÉCRIRE DANS CE CADRE

Q12

```
def Vmax(P,V,Pmax) :
    P2,V2=Tri2(P,V)
    SP=0
    SV=0
    i=0
    while ..........................................................
        SP=SP+P2[i]
        SV=SV+V2[i]
        i=i+1
    return ..........................................................
```

Q13

Profit obtenu avec cette méthode :

.....

.....

.....

.....

.....

.....

.....

.....

Commentaire :

.....

.....

Q14

Justification pour $i = 0$:

.....

.....

Justification pour $i > 0$ et $p_i > \omega$:

.....

.....

.....

Justification pour $i > 0$ et $p_i \leq \omega$:

.....

.....

.....

.....

.....

Q15

Justification de la terminaison : .....

.....

.....

.....

.....

Q16

```
def Max(a,b) :
```

Q17

```
def recur(P,V,i,w) :
     if i==0:
          return ..................................................
     if P[i-1]>w:
         return recur(P,V,i-1,w)
     else :
.....................................................................
```

Q18

.....

.....

.....

.....

Q19

.....

.....

.....

CONCOURS COMMUN INP

Numéro d'inscription

Numéro de table

Né(e) le

Nom :

Prénom :

Emplacement QR Code

Filière : TSI

Session : 2021

Épreuve de : INFORMATIQUE

Consignes

- *Remplir soigneusement l'en-tête de chaque feuille avant de commencer à composer*
- *Rédiger avec un stylo non effaçable bleu ou noir*
- *Ne rien écrire dans les marges (gauche et droite)*
- *Numéroter chaque page (cadre en bas à droite)*
- *Placer les feuilles A3 ouvertes, dans le même sens et dans l'ordre*

Q20

```
def recur2(P,V,i,w,Memoire) :
    if i==0:
        return 0
    if Memoire[i][w]>-1:
        return ..........................................................
    if P[i-1]>w:
        Memoire[i][w]=recur2(P,V,i-1,w,Memoire)
        return Memoire[i][w]
    else:
        if Memoire[i-1][w]==-1:
            Memoire[i-1][w]= ....................................
        if ..........................................................
            Memoire[i-1][w-P[i-1]]=recur2(P,V,i-1,w-P[i-1],Memoire)
    a=max(Memoire[i-1][w],V[i-1]+Memoire[i-1][w-P[i-1]])
    Memoire[i][w]=a
    return ..........................................................
```

Q21

.....

.....

NE RIEN ÉCRIRE DANS CE CADRE

Q22

. . . . . .

. . . . . .

. . . . . .

Q23

. . . . . .

. . . . . .

. . . . . .

Q24

. . . . . .

. . . . . .

. . . . . .

. . . . . .

Q25

. . . . . .

. . . . . .

. . . . . .

. . . . . .

. . . . . .

. . . . . .

Q26

.....

.....

.....

.....

Q27

```
def Identifiant2(Bin) :
```

Q28

.....

.....

.....

.....

.....

Travailler les chapitres de ce sujet

Prepa Help suit le programme de TSI chapitre par chapitre : cours détaillé, méthodes, pièges, exercices et problèmes corrigés pas à pas, tirés des concours. L'essai gratuit donne accès aux chapitres offerts de ta filière, sans carte bancaire.

Chapitres liés à ce sujet

- Bases de Python : expressions, types, instructions, fonctions
- Récursivité
- Architecture de l'ordinateur et représentation des nombres

Ouvrir l'essai gratuit

Scanne le code ou clique sur le bouton.

prepahelp.com/#essai

Une question sur le concours ou sur ta filière : **écrire sur WhatsApp**

Les autres années et épreuves : prepahelp.com/annales/ccinp/tsi