

LINGE1225 : Programmation en économie et gestion

Test semaine 9

Marco Saerens

1

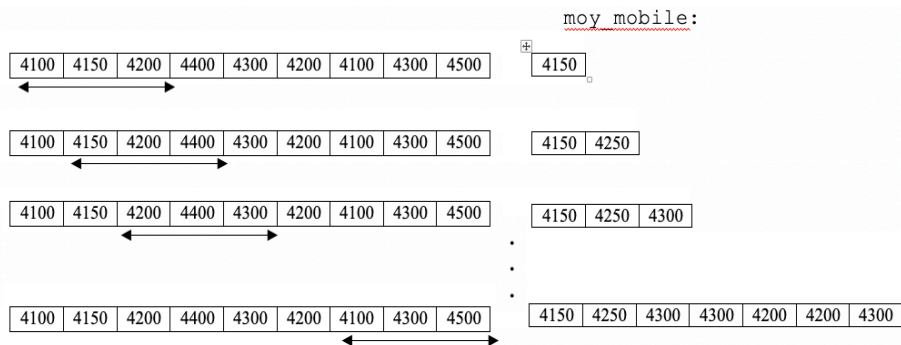
Énoncé du problème

2

Fonction 1: calculer une moyenne mobile

Contexte :

Calculer la moyenne mobile d'une série temporelle financière sur base d'un vecteur data et d'une fenêtre k (3 dans l'exemple).



Décomposition du problème

Décomposition du problème

Décomposer le problème en sous-problèmes

Notre programme peut se décomposer en sous-problèmes :

- Définir la fonction
- Tester si le nombre d'observations $\geq k$
- Initialiser les variables (indice i , liste `moy_mobile`)
- Répéter le calcul de la moyenne: `while i < len(data)`
 - Calculer la somme des k dernières observations
 - Ajouter la moyenne (somme/k) à la liste `moy_mobile`
 - Incrémenter i
- Renvoyer la liste `moy_mobile`

Décomposition du problème

Première partie

- Définir la fonction
`def moyenne_mobile02(data,k):`
- Tester si le nombre d'observations $\geq k$
`if len(data) < k: return []` # pas assez d'observations
- Initialiser les variables (indice i , liste `moy_mobile`)
`moy_mobile = []`
`i = k-1`

Décomposition du problème

Seconde partie

- Répéter le calcul de la moyenne
while i < len(data)
 - Calculer la somme des k dernières observations
sum = 0.0
for j in range(0,k):
sum = sum + data[i - j]
 - Ajouter la moyenne (somme/k) à la liste moy_mobile
moy_mobile.append(sum/k)
 - Incrémenter i
i = i + 1

7

Décomposition du problème

Troisième partie

- Renvoyer la liste moy_mobile
return moy_mobile

8

Fonction 2: calculer la longueur de la plus longue séquence croissante

Contexte :

Calculer la longueur de la plus longue séquence croissante dans la série temporelle.

4150	4250	4300	4300	4200	4200	4300
------	------	------	------	------	------	------

9

Décomposition du problème

10

Décomposition du problème

Décomposer le problème en sous-problèmes

Notre programme peut se décomposer en sous-problèmes :

- Définir la fonction
- Tester si le nombre d'observations $\geq k$
- Calculer les moyennes mobiles
- Initialiser les variables (longueur maximale maximum, longueur courante cpt)
- Répéter le calcul du maximum: for i in range(0, len(moy_mobile) - 1)
 - Calculer la longueur courante
 - Vérifier si la longueur courante excède la plus longue (update du maximum)
- Renvoyer la longueur maximale

Décomposition du problème

Première partie

- Définir la fonction
plus_longue_periode_croissante01(data,k):
- Tester si le nombre d'observations $\geq k$
if len(data) < k: return -1 # pas assez d'observations
- Calculer les moyennes mobiles
moy_mobile = moyenne_mobile01(data,k) # calcule la moyenne mobile
- Initialiser les variables (longueur maximale long, longueur courante cpt)
maximum, cpt = 0, 1

Décomposition du problème

Seconde partie

- Répéter le calcul du maximum
for i in range(0, len(moy_mobile) - 1):
 - Calculer la longueur courante à l'aide d'un compteur
if moy_mobile[i+1] > moy_mobile[i]: # si strictement croissant
cpt = cpt + 1 # on augmente le compteur
else: cpt = 1 # remettre le compteur à un si pas strictement croissant
 - Vérifier si la longueur courante excède la plus longue (update du maximum)
if cpt > maximum: # si la longueur courante est > que la longueur maximale
maximum = cpt # mettre à jour la longueur maximale

13

Décomposition du problème

Troisième partie

- Renvoyer la longueur maximale
return maximum

14