

exercice_echelle

April 14, 2021

1 1A.algo - Calculer le nombre de façons de monter une échelle.

Une grenouille monte une échelle. Elle peut faire des bonds de un ou deux barreaux. Combien a-t-elle de façons de monter une échelle de treize barreaux ? Notion abordée : fonction récursive.

```
[1]: from jyquickhelper import add_notebook_menu
      add_notebook_menu()
```

```
[1]: <IPython.core.display.HTML object>
```

1.1 Enoncés

1.1.1 Enoncé 1

On suppose qu'une grenouille peut monter une échelle de 13 barreaux avec des sauts de 1 ou 2 barreaux. Combien a-t-elle de façons de monter les 13 premiers barreaux ?

1.1.2 Enoncé 2

Le problème est presque le même excepté que l'épaisseur des barreaux est plus ou moins grande. L'épaisseur du barreau i est un entier $e_i > 0$. Arrivée au barreau i , la grenouille peut faire un bond de k barreaux avec $k \leq e_i$. La grenouille commence toujours au barreau.

1.1.3 Enoncé 3

On considère la grenouille de l'énoncé 2 mais on souhaite connaître le chemin pour lequel la grenouille a fait le moins de bonds possible.

1.2 Solutions

1.2.1 Solution au problème 1

L'idée de départ consiste à considérer le problème par récurrence. D'où la grenouille peut-elle venir si elle est au barreau n ? La réponse est assez simple : - elle a pu venir du barreau $n - 1$ avec un saut de un barreau, - elle a pu venir du barreau $n - 2$ avec un saut de deux barreaux

On obtient la formule de récurrence suivante : $f(n) = f(n - 1) + f(n - 2)$. C'est une [suite de Fibonacci](#).

```
[2]: def grenouille(n):
      a = 1 # 1 façon d'arriver au barreau 0
      b = 1 # 1 façon d'arriver au barreau 1
      i = 1
      while i < n :
          a,b = b, a + b
          i += 1
```

```

return b

for i in range(0,14) :
    print ( "barreau {0} : {1} façons de monter".format( i, grenouille(i) ) )

```

```

barreau 0 : 1 façons de monter
barreau 1 : 1 façons de monter
barreau 2 : 2 façons de monter
barreau 3 : 3 façons de monter
barreau 4 : 5 façons de monter
barreau 5 : 8 façons de monter
barreau 6 : 13 façons de monter
barreau 7 : 21 façons de monter
barreau 8 : 34 façons de monter
barreau 9 : 55 façons de monter
barreau 10 : 89 façons de monter
barreau 11 : 144 façons de monter
barreau 12 : 233 façons de monter
barreau 13 : 377 façons de monter

```

1.2.2 Solution au problème 2

La solution au second problème est presque la même. L'idée de départ consiste toujours à considérer le problème par récurrence. D'où la grenouille peut-elle venir si elle est au barreau n ? La réponse est assez simple : - elle a pu venir du barreau $n-1$ avec un saut de un barreau si $e_{n-1} \geq 1$ - elle a pu venir du barreau $n-2$ avec un saut de deux barreaux si $e_{n-2} \geq 2$ - elle a pu venir du barreau $n-k$ avec un saut de deux barreaux si $e_{n-k} \geq k$

```

[3]: def grenouille2(n, barreaux):
    nb = [ 1,1] # 1 façon d'arriver au barreau 0 ou 1
    while len(nb) < n:
        s = 0
        i = len(nb)
        for k in range(0, i):
            if barreaux[i-k-1] >= k+1:
                s += nb[i-k-1]
        nb.append(s)
        i += 1
    return nb[-1]

# on regarde si on obtient le même résultat que précédemment
g1 = [ 2 for i in range(0,14) ]
for i in range(0,14) :
    print ( "barreau {0} : {1} = {2} façons de monter".format( i, grenouille(i),
↳grenouille2(i,g1) ) )

```

```

barreau 0 : 1 = 1 façons de monter
barreau 1 : 1 = 1 façons de monter
barreau 2 : 2 = 1 façons de monter
barreau 3 : 3 = 2 façons de monter
barreau 4 : 5 = 3 façons de monter
barreau 5 : 8 = 5 façons de monter
barreau 6 : 13 = 8 façons de monter
barreau 7 : 21 = 13 façons de monter

```

```

barreau 8 : 34 = 21 façons de monter
barreau 9 : 55 = 34 façons de monter
barreau 10 : 89 = 55 façons de monter
barreau 11 : 144 = 89 façons de monter
barreau 12 : 233 = 144 façons de monter
barreau 13 : 377 = 233 façons de monter

```

La solution est légèrement différente car contrairement au premier énoncé, la grenouille ne peut aller directement au barreau 2. On essaye avec un autre exemple trivial mais qui permet de s'assurer que la fonction retourne bien le résultat voulu.

```

[4]: g2 = [ 1 for i in range(0,14) ]
      g2[4] = 3
      for i in range(0,14) :
          print ( "barreau {0} : {1} façons de monter".format( i, grenouille2(i,g2) ) )

```

```

barreau 0 : 1 façons de monter
barreau 1 : 1 façons de monter
barreau 2 : 1 façons de monter
barreau 3 : 1 façons de monter
barreau 4 : 1 façons de monter
barreau 5 : 1 façons de monter
barreau 6 : 1 façons de monter
barreau 7 : 2 façons de monter
barreau 8 : 3 façons de monter
barreau 9 : 3 façons de monter
barreau 10 : 3 façons de monter
barreau 11 : 3 façons de monter
barreau 12 : 3 façons de monter
barreau 13 : 3 façons de monter

```

1.2.3 Solution au problème 3

Pour ce problème, la solution est quasi identique excepté qu'au lieu de sommer les nombres de chemins, on souhaite garder le nombre de bonds du chemin qui en contient le moins.

```

[5]: def grenouille3(n, barreaux):
      nb = [ 1,1] # 1 façon d'arriver au barreau 0 ou 1
      while len(nb) < n:
          s = 0
          i = len(nb)
          for k in range(0, i):
              if barreaux[i-k-1] >= k+1 and (s == 0 or nb[i-k-1] < s) :
                  # on prend le minimum
                  s = nb[i-k-1]
          nb.append(s+1)
          i += 1
      return nb[-1]

      g1 = [ 2 for i in range(0,14) ]
      for i in range(0,14) :
          print ( "barreau {0} : {1} bonds pour le chemin qui en contient le moins".format(
              ↪ i, grenouille3(i,g1) ) )

```

barreau 0 : 1 bonds pour le chemin qui en contient le moins
barreau 1 : 1 bonds pour le chemin qui en contient le moins
barreau 2 : 1 bonds pour le chemin qui en contient le moins
barreau 3 : 2 bonds pour le chemin qui en contient le moins
barreau 4 : 2 bonds pour le chemin qui en contient le moins
barreau 5 : 3 bonds pour le chemin qui en contient le moins
barreau 6 : 3 bonds pour le chemin qui en contient le moins
barreau 7 : 4 bonds pour le chemin qui en contient le moins
barreau 8 : 4 bonds pour le chemin qui en contient le moins
barreau 9 : 5 bonds pour le chemin qui en contient le moins
barreau 10 : 5 bonds pour le chemin qui en contient le moins
barreau 11 : 6 bonds pour le chemin qui en contient le moins
barreau 12 : 6 bonds pour le chemin qui en contient le moins
barreau 13 : 7 bonds pour le chemin qui en contient le moins

Si on veut conserver le chemin précis qui contient le moins de bonds possibles, il suffit de mémoriser le barreau précédent i que la grenouille a emprunté pour aller en un minimum de bonds au barreau n : on conserve en mémoire le **prédécesseur**. Pour construire le meilleur chemin, on remonte de prédécesseur en prédécesseur.

[6] :