Branch: master ▾ 03MAIR---Algoritmos-de-Optimizacion / AG1 / AG1_Leonardo_Pacheco.ipynb

Find file Copy path

pachecoleonardo Creado mediante Colaboratory f35cfa0 a minute ago

1 contributor

232 lines (232 sloc) | 6.17 KB

Raw Blame History

In [0]:

AG - Actividad Guiada 1

Leonardo Pacheco

https://github.com/pachecoleonardo/03MAIR---Algoritmos-de-Optimizacion/tree/master/AG1

In [4]:
```
#quick_sort

A = [9187, 244, 4054, 9222, 8373, 4993, 5265, 5470, 4519, 7182, 2035, 3506, 4337, 7580, 2554, 2
824, 8357, 4447, 7379]

def quick_sort(A):
  if len(A) == 1:
    return A
  if len(A) == 2:
    return [min(A), max(A)]

  IZQ=[]
  DER=[]

  pivote = (A[0] + A[1] + A[2])/3

  for i in A:
    if i <= pivote :
      IZQ.append(i)
    else:
      DER.append(i)

  return quick_sort(IZQ) + quick_sort(DER)

quick_sort(A)
```

Out[4]:
```
[244,
 2035,
 2554,
 2824
```

```
3506,
4054,
4337,
4447,
4519,
4993,
5265,
5470,
7182,
7379,
7580,
8357,
8373,
9187,
9222]
```

In [4]:
```
SISTEMA = [25, 10, 5, 1]

def cambio_monedas(CANTIDAD, SISTEMA):
  SOLUCION =[0 for i in range(len(SISTEMA))]

  VALOR_ACUMULADO = 0

  for i in range(len(SISTEMA)):
    monedas = int((CANTIDAD - VALOR_ACUMULADO)/SISTEMA[i])

    SOLUCION[i] = monedas

    VALOR_ACUMULADO += monedas*SISTEMA[i]

    if CANTIDAD == VALOR_ACUMULADO:
      return SOLUCION

cambio_monedas(99, SISTEMA)
```

Out[4]:
```
[3, 2, 0, 4]
```

In [0]:
```
N=4

solucion = [0 for i in range(N)]
```

```
solucion = [0 for i in range(N)]

etapa=0

def es_prometedora(SOLUCION,etapa):
  for i in range(etapa+1):
    if SOLUCION.count(SOLUCION[i]) > 1:   return False

    #Verificar las diagonales
    for j in range (i+1, etapa +1):
      if abs(i-j) == abs (SOLUCION[i]-SOLUCION[j]): return False
  return True

def escribe(S):
  n = len (S)
  for x in range(n):
    print("")
    for i in range(n):
      if solucion[i] == x+1:
        print(" X " , end="")
      else:
        print(" - ", end="")


def reinas(N, solucion, etapa): #Tamaño del tablero

  for i in range(1,N+1):
    solucion[etapa] = i

  #print(solucion)
  if es_prometedora(solucion, etapa):
    if etapa == N-1:
      print("\n\nLa solución es:")
      print(solucion)
      escribe(solucion)
    else:
      #print("Es prometedora\n#################")
      reinas(N, solucion, etapa+1)
  else:
    #print("NO PROMETEDORA\n#################")
```

```
    None

    solucion[etapa] = 0


reinas(N, solucion, etapa)
```