

Tarea Diseño de Algoritmos

Estudiante: Ignacio Lara Vidal

Sección B-2

Algoritmo:

Entrada: Casos de prueba, compuestos por dos líneas, la primera con dos números, el peso total de uvas que puede comer el usuario, y la cantidad de uvas disponibles; la segunda con el peso de cada uva.

Leer Casos de Prueba # Cada caso queda [pesoUvas, [uva1, uva2, uva3...]]

Procedimiento combUvas(uvasDisp, maxUvas, actualUvas, recorrido):

 para i = 1 hasta largo(uvasDisp):

 si uvasDisp[i] != 0:

 si actualUvas + uvasDisp[i] <= maxUvas:

 actualUvas = actualUvas + uvasDisp[i]

 recorrido.agregar(i)

 uvasDisp[i] = 0

 combUvas(uvasDisp, maxUvas, actualUvas, recorrido)

 si no:

 si largo(recorrido) > 1:

 recorridosTotal.agregar(recorrido)

Para i = 1 hasta largo(Casos):

 recorridosTotal = [] # Lista vacía

 combUvas(Casos[i][listaUvas], Casos[i][pesoUvas], 0, [])

#El último parámetro es una lista vacía

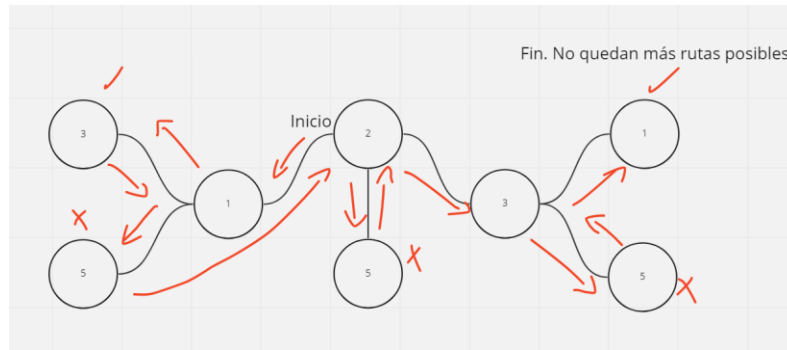
 eliminarRepetidos(recorridosTotal) # Elimina todos los recorridos que
 tengan los mismos números

 mostrar Largo(recorridosTotal)

Idea del algoritmo:

Utilizando el método de backtracking, lo que hacemos es una función recursiva, que en palabras sencillas, tomará cada uva, y comprobará con el resto de uvas si es viable, luego con cada combinación hará lo mismo con el resto de uvas. Con una imagen quedará más claro.

Suponiendo que hay 4 uvas distintas, de peso [1, 2, 3, 5], nos quedaría un esquema así:



Posteriormente, se realizaría el mismo proceso, iniciando en las otras 3 uvas restantes, agregando cada ruta obtenida a un listado. Finalmente, se filtran las rutas duplicadas (Que cruzan los mismos “nodos”), y se muestra la cantidad restante de rutas en pantalla.

Respecto al tiempo del algoritmo:

Siendo j = cantidad de entradas y n la cantidad de uvas

$$T(n) = j * (3 + (8*n)*(8*(n-1))*(8*(n-2))...*(8*(n-n)))$$

$$O(n) = j * n^{(n-1)}$$

-> Asumiendo que $j = 1$, siendo un caso por ejecución.

$$O(n) = n^{(n-1)}$$

De esto, la verdad es que lo único mencionable es que su eficiencia es mínima, y resulta ser muy poco útil si se deseara utilizar en casos con números muy grandes, de hecho, su uso debería ser evitado para cualquier propósito no educacional.