

# Actividad 1.5 Algoritmos de branch and bound

Victor Misael Escalante Alvarado

A01741176

Algoritmo de branch-and-bound para coin-collecting.

Screenshot del algoritmo funcionando

This screenshot shows the initial setup of the branch-and-bound algorithm. The code defines a recursive function `recursion` that takes a list of coins and a target value. It uses a dictionary `visited` to avoid revisiting states. The function returns a list of coins that sum up to the target value. The main part of the code initializes the `visited` dictionary and calls the `recursion` function with the initial state.

```
def recursion(coins, target, visited):
    if target == 0:
        return []
    if target < 0:
        return None
    if target in visited:
        return None
    visited[target] = True
    for coin in coins:
        result = recursion(coins, target - coin, visited)
        if result is not None:
            return result + [coin]
    return None
```

This screenshot shows the implementation of the branch-and-bound algorithm. The code defines a recursive function `recursion` that takes a list of coins and a target value. It uses a dictionary `visited` to avoid revisiting states. The function returns a list of coins that sum up to the target value. The main part of the code initializes the `visited` dictionary and calls the `recursion` function with the initial state.

```
def recursion(coins, target, visited):
    if target == 0:
        return []
    if target < 0:
        return None
    if target in visited:
        return None
    visited[target] = True
    for coin in coins:
        result = recursion(coins, target - coin, visited)
        if result is not None:
            return result + [coin]
    return None
```

Como calcularon el upper bound ?

El upper bound se calculó por medio de tomar el maximo de monedas que quedaban en las columnas y filas de la celda donde iniciaria la solución hija y de ahi se decide la prioridad para esa solución

Enlace a repli : <https://replit.com/join/pwsxlzqdt-vmisa>