# HW6

Xinhao Luo

Tuesday 12$^{\text{th}}$ May, 2020

# 1 DPV Problem 6.18

Define l as len(n), where n a list of denominations.
Define sub-problem arr[l][v] where each arr[x][y] means the using number in range of [1...x], if weight y can be reached.

```
initialize arr[l][v] as false
initialize arr[0][v] and arr[l][0] as true
for i in range(1, l + 1):
    for j in range(1, v + 1):
            arr[i][j] = arr[i - 1][j - n[i - 1]] or arr[i - 1][j]

return arr[l][v]
```

The for-loop indicate that the time complexity for this algorithm is $O(nv)$

For each step, we will check if the current target j can fit in the given denomination i. If it can, we will find if v without this number can be reached. If it can't, we just keep the result from the previous calculation (i - 1).

## 2 DPV Problem 6.19

Define l as len(n), where n a list of denominations.
Define sub-problem arr[v], where each arr[x] means the minimum number of coins needs to use to reach target x.

```
initialize arr[v] as Infinity
initialize arr[0] = 0
for i in range(1, v + 1):
    for j in range(l):
        if n[i - 1] <= i:
            arr[i] = min(arr[i - n[j]] + 1, arr[i])

return arr[v] <= v
```

The for-loop indicate that the time complexity for this algorithm is $O(nv)$

For each step, we will check if a coin can fit in the target range. If it can, we will find the minimum number of coin needed to reach this target with it or without this denomination; and keep the smaller result.

# 3 DPV Problem 7.18

for part (a), show how to use the original max-flow problem to solve this variation; and for part (c), show how to use linear programming tosolve this.

a) Set new vertices S, T where S connects to all sources, and T is connected by all sink. All edge start from S or end at T no capacity limit.
Now the whole graph has only one source and one sink. If we can calculate the max flow start from S to T then we will have the max flow for the original graph.
The time complexity should be the same as $O(m)$, where m is the max flow of the graph.
The result is valid as the additional edges has unlimited flow while the rest of the graph keeps the same. The threshold are the edges' capacity within the original graph.

c) By default in LP model, we have 0 as the lower bound.
When we defining the LP model, we simply add another constraint that if the edge has lower bound, we show that constraint on the function.
The time complexity should remain the same as the LP problem.