

Data Structure and Advanced Programming

Homework 3

第一題

(a) Catalan Number with Recursive method

```
int catalanRecursive(int n)
{
    if (n <= 0)
        return 0;
    else if (n == 1)
        return 1;
    else
        return 2 * (2 * n - 1) * catalanRecursive(n - 1) / (n + 1);
}
```

(b) Knapsack Problem

```
1  input: w[ 1, ..., n], v[ 1, ..., n], V, B
2
3  define w = sum(w[i] * x[i]) where x[i] is in {0, 1}
4  define P(n, B) is the solution to the problem with n items, each with w[i] and v[i],
5      and the knapsack capacity B
6
7  for w = 0 to B
8      P(0, w) <- 0
9  for i = 0 to n
10     for w = 0 to B
11         if w[i] > w
12             P(i, w) <- P(i - 1, w)
13         else
14             P(i, w) = max { P(i - 1, w), v[i] + P(i - 1, w - w[i]) }
15 if P(n, B) > V
16     return true
17 else
18     return false
```

(c) Pseudocode for Augmenting Path

```

1  input: number of nodes n, edges m, an adjacency matrix adj[n][n]
2  define Q <- vector, s <- s[0], t <- s[n]
3  // below follows the functions in c++ library <vector>
4  // in the queue/vector are Paths practiced in class
5
6  Q.push_back(s)
7  while(!Q.empty() && terminationNotFound)
8      curPath <- Q.front()
9      curNode <- Q.front().getTail()
10     Q.erase(Q.begin())
11
12     isLoop <- false
13     vector<Path> newPaths <- getNextWithResCap(curNode)
14
15     for i <- 1 to curPath.size()
16         for j <- 1 to curPath.size()
17             if curPath.node[i] == curPath.node[j]
18                 isLoop <- true
19
20     if isLoop
21         continue;
22     else
23         for i <- 0 to newPaths.size()
24             if(newPaths[i].getTail() == t)
25                 return newPaths[i]
26             else
27                 Q.push_back(newPaths[i])
28
29 return -1

```

(d) Revised Maximum Flow Problem

(i) Multiple Sources and Terminations

新增兩個假想的起點 s' 與終點 t' 來連接多個起點 $\{s\}_{i=1}^3$ 與終點 $\{t\}_{i=1}^2$ ，並且定義 s' 與 $\{s\}_{i=1}^3$ 之間及 t' 與 $\{t\}_{i=1}^2$ 之間的邊的容量都是無限大，就可以應用原本的程式了。

(ii) Vertices with Capacities

把每個點 v_i 都分成兩個點 v_{in} 跟 v_{out} ，並且以有向邊從 v_{in} 指向 v_{out} ，定義這條邊的容量是原先指定點的容量，就可以應用原程式。