DSAP Homework3

B05703100 財金三 郭仲嘉

第一題

(a)

#include <iostream>

using namespace std;

long long cat(int num)

{

if (num == 0 or num == 1) return 1;

else

{

long long result = 0;

for (int k = 0; k < num; k++)

result += ( cat(k) \* cat(num - 1 - k) );

return result;

}

}

int main()

{

int num;

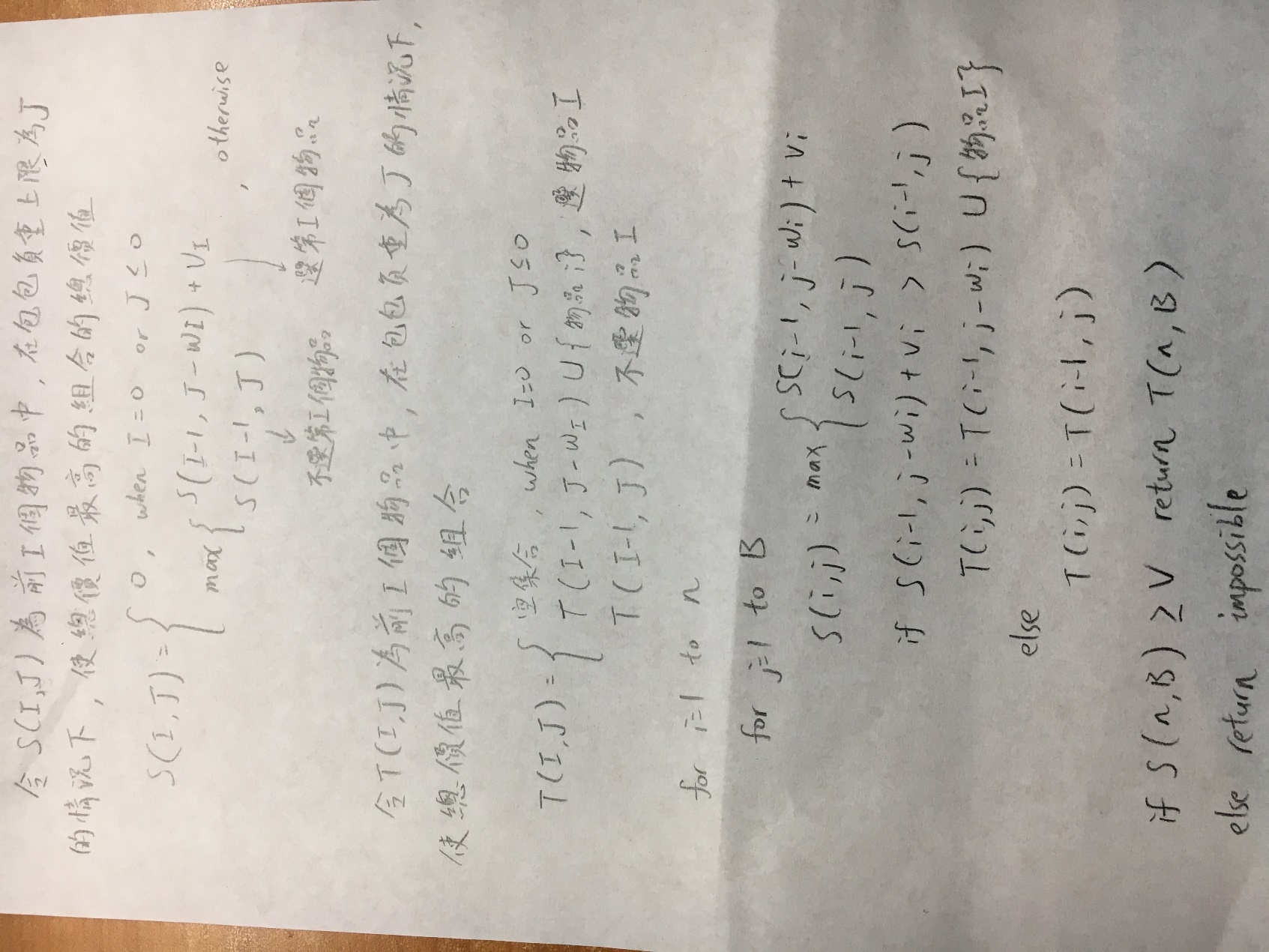
cin >> num;

cout << cat(num);

return 0;

}

(b)



(c)

Queue<Path> q

Node s, t

for all nextPath of s

if capacity(nextPath == 0) continue

else q.enqueue(nextPath)

while (! q.empty())

for all nextPath of q.front()

if nextPath reaches t, return nextPath

if capacity(nextPath) == 0, continue

if nextPath makes loop, continue

capacity(nextPath) = min(capacity(nextPath), capacity(q.front())

q.enqueue(nextPath)

q.dequeue()

return Path not Found

(d)

(i)

從v1開始用原演算法，但搜尋到t1或t2都可視為找到augmenting path。

v1搜尋完畢後再以同樣規則搜尋完v2、v3。

(ii)

紀錄每個地點的Ui。搜尋到augmenting path時，紀錄增加此path每個地點會減少或增加(反方向流經)的流量。而搜尋時path的maximum flow現為：原Path的流量、下一條邊的剩餘流量、及下一個地點的剩餘流量三者的最小值。

原演算法中下一條邊的剩餘流量為0時，或加入下一個地點即形成圈圈時就不排入搜尋的Queue中。另加入一條件，下一個地點的剩餘流量為0時就不排入搜尋的Queue。