

### Раздел «Алгоритмы» . MaxFlowPushRelabelCPP :

#### Программа поиска максимального потока методом проталкивания предпотока

- "Максимальный поток" — формулировка задачи и теория

Метод проталкивания предпотока (Push-Relabel или Lift-To-Front):

```
/******  
Simple implementation of the push-relabel algorithm  
for the maximal network flow.  
Input format:  
n s t - number of nodes, source and sink  
then follow c[i][j], if i == j then number is omitted.  
See read() function.  
******/  
  
#include <iostream>  
#include <fstream>  
#include <cstdlib>  
#include <algorithm>  
#include <list>  
  
using namespace std;  
const int N = 2000;  
  
int e[N],c[N][N],h[N];  
int n,s,t;  
  
void push(int u,int v)  
{  
    int f = min(e[u],c[u][v]);  
    e[u] -= f; e[v] += f;  
    c[u][v] -= f; c[v][u] += f;  
}  
  
void lift(int u)  
{  
    int min = 3 * n + 1;  
  
    for (int i = 0;i < n;i++)  
        if (c[u][i] && (h[i] < min))  
            min = h[i];  
    h[u] = min + 1;  
}  
  
void discharge(int u)  
{  
    int v = 0;  
    while (e[u] > 0)  
    {  
        if (c[u][v] && h[u] == h[v] + 1)  
        {  
            push(u,v); v = 0; continue;  
        }  
        v++;  
        if (v == n)  
        {  
            lift(u); v = 0;  
        }  
    }  
}
```

Поиск

Поиск

Раздел  
«Алгоритмы»

Главная

Форум

Ссылки

El Judge

Инструменты:

Поиск

Изменения

Index

Статистика

Разделы

Информация

Алгоритмы

Язык Си

Язык Ruby

Язык

Ассемблера

El Judge

Парадигмы

Образование

Сети

Objective C

Logon>>

```
    }  
}  
  
void read()  
{  
    cin >> n >> s >> t;  
  
    for (int i = 0; i < n; i++)  
        for (int j = 0; j < n; j++)  
        {  
            if (i == j)  
                continue;  
            cin >> c[i][j];  
        }  
}  
  
void init()  
{  
    read();  
    for (int i = 0; i < n; i++)  
    {  
        if (i == s)  
            continue;  
        e[i] = c[s][i]; c[i][s] += c[s][i];  
    }  
    h[s] = n;  
}  
  
int main(int argc, char *argv[])  
{  
    list<int> l;  
    list<int>::iterator cur;  
    int old;  
  
    init();  
  
    for (int i = 0; i < n; i++)  
        if (i != s && i != t)  
            l.push_front(i);  
    cur = l.begin();  
  
    while (cur != l.end())  
    {  
        old = h[*cur];  
        discharge(*cur);  
        if (h[*cur] != old)  
        {  
            l.push_front(*cur); l.erase(cur); cur = l.begin();  
        }  
        cur++;  
    }  
    cout << e[t];  
    return 0;  
}
```

Copyright © 2003–2022 by the contributing authors.