# 网络流sap模板

*001* #include <iostream>  
*002* #include <cstdio>  
*003* #include <cstring>  
*004* #include <algorithm>  
005 #define OP(i) (((i) - (pool))^1)  
*006* **using** **namespace** std;  
*007* **class** sap  
*008* {  
*009* **private**:  
010     **const** **static** **int** V = 6000,E = 1000000,INF = 100000000;  
*011*     **int** dis[V],numdis[V];  
*012*     **struct** edge  
*013*     {  
*014*         **int** v,cap;  
015         edge \*nxt;  
*016*     }pool[E],\*g[V],\*pp;  
*017*     **void** bfs(**int** ii,**int** n)  
*018*     {  
*019*         **int** que[V],tail = 0;  
020         **bool** vst[V] = {0};  
*021*         memset(numdis,0,**sizeof**(numdis));  
*022*         fill(dis,dis + n,n);  
*023*         dis[ii] = 0;vst[ii] = 1;que[0] = ii;  
*024*         **for** (**int** j = 0;j <= tail;j++)  
025         {  
*026*             **int** tmp = que[j % n];  
*027*             **for** (edge \*i = g[tmp];i != NULL;i = i->nxt)  
*028*             {  
*029*                 **if** (pool[OP(i)].cap > 0 && !vst[i->v])  
030                 {  
*031*                     tail++;  
*032*                     vst[i->v] = 1;  
*033*                     que[tail % n] = i->v;  
*034*                     dis[i->v] = dis[tmp] + 1;  
035                     numdis[dis[i->v]]++;  
*036*                 }  
*037*             }  
*038*         }  
*039*     }  
040 **public**:  
*041*     **int** maxflowsap(**int** n,**int** s,**int** t)  
*042*     {  
*043*         bfs(t,n);  
*044*         **int** v = s,skip,pre[V],ans = 0;  
045         edge \*e[V],\*pree[V];  
*046*         memcpy(e,g,**sizeof**(g));  
*047*         **while** (dis[s] < n)  
*048*         {  
*049*             skip = 0;  
050             **while** (e[v])  
*051*             {  
*052*                 **if** (e[v]->cap > 0 && dis[v] == dis[e[v]->v] + 1)  
*053*                 {  
*054*                     pre[e[v]->v] = v;  
055                     pree[e[v]->v] = e[v];  
*056*                     v = e[v]->v;  
*057*                     **if** (v == t)  
*058*                     {  
*059*                         **int** minf = INF;  
060                         **for** (**int** i = t;i != s;i = pre[i])  
*061*                             minf = min(minf,pree[i]->cap);  
*062*                         ans += minf;  
*063*                         **for** (**int** i = t;i != s;i = pre[i])  
*064*                         {  
065                             pree[i]->cap -= minf;  
*066*                             pool[OP(pree[i])].cap += minf;  
*067*                         }  
*068*                         v = s;  
*069*                         skip = 1;  
070                         **break**;  
*071*                     }  
*072*                 }  
*073*                 **else**  
*074*                     e[v] = e[v]->nxt;  
075             }  
*076*             **if** (!skip)  
*077*             {  
*078*                 **int** mindis = INF;  
*079*                 numdis[dis[v]]--;  
080                 **if** (!numdis[dis[v]])  
*081*                     **break**;  
*082*                 **for** (edge \*i = g[v];i != NULL;i = i->nxt)  
*083*                     **if** (i->cap > 0)  
*084*                         mindis = min(mindis,dis[i->v] + 1);  
085                 **if** (mindis == INF)  
*086*                     dis[v] = n;  
*087*                 **else**  
*088*                     dis[v] = mindis;  
*089*                 numdis[dis[v]]++;  
090                 e[v] = g[v];  
*091*                 **if** (v != s)  
*092*                     v = pre[v];  
*093*             }  
*094*         }  
095         **return** ans;  
*096*     }  
*097*     **void** firststart()  
*098*     {  
*099*         pp = pool;  
100         memset(g,0,**sizeof**(g));  
*101*     }  
*102*     **void** addedge(**int** i,**int** j,**int** cap)  
*103*     {  
*104*         pp->v = j;  
105         pp->cap = cap;  
*106*         pp->nxt = g[i];  
*107*         g[i] = pp++;  
*108*     }  
*109* }maxflow;