MC202 — ESTRUTURAS DE DADOS

Laboratório 11 — Grafos

Tarefa

Em um sistema de tubulações, é feito o monitoramento da quantidade de água que entra e sai. Pode haver diversos pontos de entrada e saída. Se a soma de tudo que entra for maior do que a soma de tudo o que sai, está havendo desperdício nesse sistema. Sua tarefa é calcular esse desperdício.

Para representar um sistema de tubulações, será usado um grafo direcionado, onde os tubos são representados por arestas com uma dada capacidade e as conexões entre os tubos são representadas por vértices. Em cada tubo correspondente a uma aresta (u, v) há um medidor que calcula a vazão $f(u, v)$ de água que passa pelo tubo. Se $c(u, v)$ for a capacidade do tubo, então:

$$f(u, v) < c(u, v)$$

O conjunto de vértices S indica as fontes de água que abastecem o sistema. Para cada fonte $s \in S$, a quantidade de água produzida em s é dada pela soma do fluxo na arestas que começam em s menos a soma do fluxo das arestas que terminam em s. A quantidade total de água produzida no sistema é denotada por F_{in}.

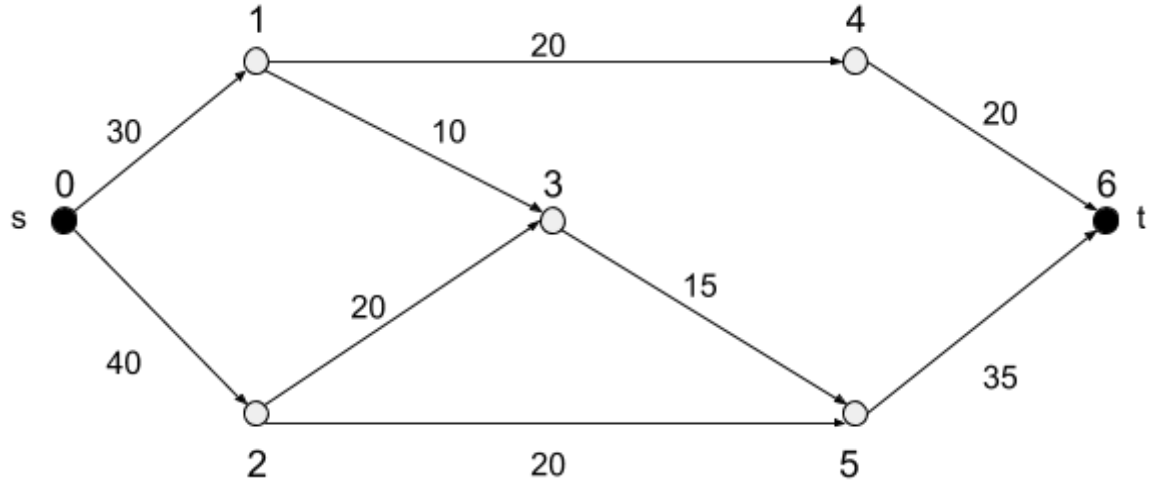
Já o conjunto de vértices T indica os sorvedouros que consomem água do sistema. Para cada sorvedouro $t \in T$, a quantidade de água consumida por t é dada pela soma do fluxo das arestas que terminam em t menos a soma do fluxo das arestas que começam em t. A quantidade total de água consumida pelo sistema é denotada por F_{out}.

Para cada um dos demais vértices v (que não estão em S nem em T), pode ser que o fluxo de saída seja menor do que o fluxo de entrada. Nesse caso, há um vazamento no vértice v correspondente a essa diferença.

Assim, o fluxo total do sistema é dado por:

$$F = F_{in} - F_{out}$$

Se F for maior que zero, há desperdício igual a F. Se for igual a zero, não há desperdício.



No grafo acima, há desperdício no vértice 3. O desperdício total é calculado como: $(30 + 40) - (20 + 35) = 15$.

Entrada

Na primeira linha, há um número de vértices, $n\,(2 \leq n \leq 4 \times 10^2)$. Os vértices são identificados por números entre 0 e $n-1$.

Na segunda linha, há um número de arestas, m $(n-1 \leq m \leq n(n+1)/2)$. Em seguida, há m linhas, cada uma com três inteiros: os dois primeiros correspondem a identificadores de vértices (u, v) que representam os extremos da aresta de u para v e o terceiro é o fluxo $f(u,v)$ que passa pela aresta.

Em seguida, há um inteiro $S\,(1 \leq S < n)$, que é o número de fontes. A linha seguinte contém os S identificadores dos vértices de entrada.

Por fim, há um número $T(1 \leq T \leq n - S)$, que é o número de sorvedouros. A linha seguinte contém os T identificadores dos vértices de saída.

Saída

A primeira linha contém o total de água desperdiçada pelo sistema, ou 0 caso não haja desperdício.

Em seguida, há uma linha para para cada vértice com vazamento, em ordem de identificador, contendo o identificador do vértice e o vazamento correspondente.

Exemplo

Entrada

```
3
2
0 1 10
1 2 10
1
0
1
2
```

Saída

```
0
```

Entrada

```
7
9
0 1 30
0 2 40
1 4 20
1 3 10
2 3 20
2 5 20
3 5 15
4 6 20
5 6 35
1
0
1
6
```

Saída

```
15
3 15
```

Critérios específicos

- Deverão ser submetidos os seguintes arquivos:

    - digrafo.c (implementação do grafo dirigido)
    - digrafo.h (interface do grafo dirigido)
    - lab11.c (programa principal cuja solução utiliza grafos)
- É obrigatório resolver usando grafos.
- Tempo máximo de execução: 5 segundos.

Testando

Para compilar usando o Makefile fornecido e verificar se a solução está correta basta seguir o exemplo abaixo.

```
make
./lab11 < arq01.in > arq01.out
diff arq01.out arq01.res
```

onde `arq01.in` é a entrada (casos de testes disponíveis no SuSy) e `arq01.out` é a saída do seu programa. O Makefile também contém uma regra para testar todos os testes de uma vez; nesse caso, basta digitar:

```
make testar_tudo
```

Isso testará o seu programa com os casos abertos. Após o prazo, os casos de teste fechados serão liberados e podem ser baixados digitando-se:

```
make baixar_fechados
```

Para ver quanto tempo seu programa demora em cada teste, digite:

```
make tempo
```

Observações gerais

No SuSy, haverá 3 tipos de tarefas com siglas diferentes para cada laboratório de programação. Todas possuirão os mesmos casos de teste. As siglas são:

1. **DRAFT**: Esta tarefa serve para testar o programa no SuSy antes de submeter a versão final. Nessa tarefa, tanto o prazo quanto o número de submissões são ilimitados, porém arquivos submetidos aqui não serão corrigidos.
2. **ENTREGA**: Esta tarefa tem limite de *uma única submissão* e serve para entregar a versão final dentro do prazo estabelecido para o laboratório. Não use essa tarefa para testar o seu programa: submeta aqui quando não for mais fazer alterações no seu programa.

3. **FORAPRAZO**: Esta tarefa tem limite de *uma única submissão* e serve para entregar a versão final após o prazo estabelecido para o laboratório, mas com nota reduzida (conforme a ementa). O envio nesta tarefa irá substituir a nota obtida na tarefa ENTREGA apenas se o aluno tiver realizado as correções sugeridas no feedback ou caso não tenha enviado anteriormente em ENTREGA.

Observações sobre SuSy:

- Versão do GCC: gcc (GCC) 4.8.5 20150623 (Red Hat 4.8.5-36)
- Flags de compilação:
  `-std=c99 -Wall -Werror -Werror=vla -pedantic-errors -g -lm`

Além das observações acima, esse laboratório será avaliado pelos critérios gerais:

- Indentação de código e outras boas práticas, tais como:
    - uso de comentários (úteis e apenas quando forem relevantes);
    - código simples e fácil de entender;
    - sem duplicidade (partes que fazem a mesma coisa).
- Organização do código:
    - tipos de dados criados pelo usuário e funções bem definidas e tão independentes quanto possível.
- Corretude do programa:
    - programa correto e implementado conforme solicitado no enunciado;
    - inicialização de variáveis sempre que for necessário;
    - dentre outros critérios.