INSTITUTO SUPERIOR TÉCNICO

Análise e Síntese de Algoritmos 2020/2021

2º Projecto

Data Limite de Entrega: 21 de Maio de 2021 (23h59)

Descrição do Problema

Em computação distribuída, o problema de atribuição de um conjunto de processos a um conjunto de processadores é um problema complexo, mesmo quando considerados apenas dois processadores, devido a *overheads* computacionais, como por exemplo: custos de execução de cada processo em cada processador (este pode ter diferentes capacidades - velocidade, cache, etc); e custos de comunicação entre os vários processos.

Considere que pretende correr um programa $P = \{p_1, \ldots, p_n\}$, constituído por um conjunto de n processos. Considere ainda que é conhecido o custo X_i de correr cada processo p_i no processador X, e o custo Y_i de correr cada processo p_i no processador Y.

Considere adicionalmente o custo de comunicação c_{ij} entre dois processos p_i e p_j, sempre que $p_i \neq p_j$. Assuma que não há custo de comunicação ($c_{ij} = 0$) entre processos que corram num mesmo processador.

Seja $P_X \subseteq P$ o conjunto de processos que correm no processador X, e $P_Y \subseteq P$ o conjunto de processos que correm no processador Y, tal que $P_X \cap P_Y = \emptyset$, proponha e implemente um algoritmo eficiente que minimize o custo total de execução do programa P, sem testar todas as combinações de atribuições possíveis.

$$min\{\Sigma_{i \in P_X} X_i + \Sigma_{i \in P_Y} Y_i + \Sigma_{(i,j) \in P_X \times P_Y} c_{ij}\}$$

Nota (2021-05-12): Considere que a soma custos de execução $(\Sigma_{i \in P_X} X_i + \Sigma_{i \in P_Y} Y_i) \in O(n)$.

Exemplo

Custos de execução de cada processo p_i

i	1	2	3	4
X_i	6	5	10	4
Y_i	4	10	3	8

Custos de comunicação c_{ij} entre dois processos (p_i,p_j)

c_{ij}	1	2	3	4
1	0	5	0	0
2	5	0	6	2
3	0	6	0	1
4	0	2	1	0

Solução de custo mínimo

$A_X = \{p_4\}$
$A_Y = \{p_1, p_2, p_3\}$

$$\begin{aligned} custo &= \Sigma_{i\in A_X} X_i + \Sigma_{i\in A_Y} Y_i + \Sigma_{(i,j)\in A_X\times A_Y} c_{ij} \\ &= X_4 + Y_1 + Y_2 + Y_3 + c_{24} + c_{34} \\ &= 4+4+10+3+2+1 \\ &= 24 \end{aligned}$$

Input

O ficheiro de entrada contém informação sobre o número de processos, bem como a informação relativa ao custos de execução e de comunicação.

O input é definido da seguinte forma:

- Uma linha contendo dois inteiros: o número n de processos ($n \geq 2$); e o número k de entradas diferentes de 0 na matriz de custos de comunicação ($k \geq 0$), apenas na diagonal superior visto que a matriz é simétrica;
- n linhas com dois inteiros com os custos de execução de p_i no processador X e Y, respectivamente;
- k linhas contendo 3 inteiros: o identificador i ($\leq n$) do processo p_i; o identificador j ($\leq n$) do processo p_j; e o custo de comunicação c_{ij}. Dado que a matriz é simétrica, $c_{ij} = c_{ji}$.

Output

O output da sua solução deverá indicar apenas um inteiro representando o valor do custo mínimo, respeitando as restrições acima.

Exemplos

input 1

```
4 4
6 4
5 10
10 3
4 8
1 2 5
2 3 6
2 4 2
3 4 1
```

output 1

```
24
```

Implementação

A implementação do projecto deverá ser feita preferencialmente usando as linguagens de programação C ou C++. Submissões nas linguagens Java/Python também serão aceites, embora fortemente desaconselhadas. Alunos que o escolham fazer devem estar cientes de que submissões em Java/Python podem não passar todos os testes mesmo implementando o algoritmo correcto, devido a aspectos de implementação e overheads da VM/interpretador.
O tempo necessário para implementar este projecto é inferior a 15 horas.

Parâmetros de compilação:

```
C++: g++ -std=c++11 -O3 -Wall file.cpp -lm
C: gcc -O3 -ansi -Wall file.c -lm
Javac: javac File.java
Java: java -Xss32m -Xmx256m -classpath . File
Python: python3 file.py
```

Submissão do Projecto

A submissão do projecto deverá incluir um relatório resumido e um ficheiro com o código fonte da solução. Informação sobre as linguagens de programação possíveis está disponível no website do sistema Mooshak. A linguagem de programação é identificada pela extensão do

ficheiro. Por exemplo, um projecto escrito em c deverá ter a extensão .c. Após a compilação, **o programa resultante deverá ler do standard input e escrever para o standard output**. Informação sobre as opções e restrições de compilação podem ser obtidas através do botão help do sistema Mooshak. O comando de compilação não deverá produzir output, caso contrário será considerado um erro de compilação.

Relatório: deverá ser submetido através do sistema Fénix no formato PDF com não mais de **2** páginas, fonte de 12pt, e 3cm de margem. O relatório deverá incluir uma descrição da solução, a análise teórica e a avaliação experimental dos resultados. O relatório deverá incluir qualquer referência que tenha sido utilizada na realização do projecto. Relatórios que não sejam entregues em formato PDF terão nota 0. Atempadamente será divulgado um template do relatório.

Código fonte: deve ser submetido através do sistema Mooshak e o relatório (em formato PDF) deverá ser submetido através do Fénix. O código fonte será avaliado automaticamente pelo sistema Mooshak (`http://acp.tecnico.ulisboa.pt/~mooshak/`). Os alunos são encorajados a submeter, tão cedo quanto possível, soluções preliminares para o sistema Mooshak e para o Fénix. Note que apenas a última submissão será considerada para efeitos de avaliação. Todas as submissões anteriores serão ignoradas: tal inclui o código fonte e o relatório.

Avaliação

O projecto deverá ser realizado em grupos de um ou dois alunos e será avaliado em duas fases. Na primeira fase, durante a submissão, cada implementação será executada num conjunto de testes, os quais representam 85% da nota final. Na segunda fase, o relatório será avaliado. A nota do relatório contribui com 15% da nota final.

Avaliação Automática

A primeira fase do projecto é avaliada automaticamente com um conjunto de testes, os quais são executados num computador com o sistema operativo **GNU/Linux**. É essencial que o código fonte compile sem erros e respeite os standards de entrada e saída indicados anteriormente. Os projectos que não respeitem os formatos especificados serão penalizados e poderão ter nota 0, caso falhem todos os testes. Os testes **não serão divulgados antes da submissão**. No entanto, todos os testes serão disponibilizados após o deadline para submissão do projecto. Além de verificar a correcção do output produzido, o ambiente de avaliação **restringe a mémoria e o tempo de execução** disponíveis. A maior parte dos testes executa o comando `diff` da forma seguinte:

```
diff output result
```

O ficheiro `result` contém o output gerado pelo executável a partir do ficheiro `input`. O

ficheiro `output` contém o output esperado. Um programa passa num teste e recebe o valor correspondente, quando o comando `diff` não reporta quaisquer diferenças (i.e., não produz qualquer output). O sistema reporta um valor entre 0 e 170.
A nota obtida na classificação automática poderá sofrer eventuais cortes caso a análise do código demonstre recurso a soluções ajustadas a inputs concretos ou outputs aleatórios/constantes.

Detecção de Cópias

A avaliação dos projectos inclui um procedimento para detecção de cópias. A submissão de um projecto implica um compromisso de que o trabalho foi realizado exclusivamente pelos alunos. A violação deste compromisso ou a tentativa de submeter código que não foi desenvolvido pelo grupo implica a reprovação na unidade curricular, para todos os alunos envolvidos (includindo os alunos que disponibilizaram o código). Qualquer tentativa de fraude, directa or indirecta, será comunicada ao Conselho Pedagógico do IST, ao coordenador de curso, e será penalizada de acordo com as regras aprovadas pela Universidade e publicadas em "Diário da República".