



SCC0216 – Modelagem Computacional em Grafos
Prof.^a Rosane Minghim

ROTEIRO DO LABORATORIO 2

Introdução

Para quem trabalha com Linux (ou outros sistemas *UNIX-like*), os chamados “sistemas de gerenciamento de pacotes” - como *yum*, *rpm* e, no caso do *Ubuntu* e outros sistemas *Debian* (e objeto de estudo deste trabalho), *apt* - são ferramentas importantes para a administração do sistema. Entre as funcionalidades oferecidas por essas ferramentas, duas se destacam: a) o *download* automático de pacotes; e b) a resolução automática de dependências. É nesta segunda funcionalidade que focaremos este exercício.

Você irá simular o processo de resolução de dependências do *apt* para instalação e desinstalação de pacotes. A entrada consistirá em uma lista de pacotes e suas dependências, seguida de uma sequência de comandos de instalação e desinstalação a serem executados. Considerando que uma dependência deve ser instalada sempre antes do pacote dependente, ao final você deverá fornecer a lista de todos os pacotes do sistema, na sequência em que foram instalados.

Entrada

O exemplo a seguir ilustra o formato de entrada:

```
pkg_A
  Depends: pkg_B
  Depends: pkg_D
pkg_C
  Depends: pkg_D
---
install pkg_A
install pkg_C
remove pkg_B
```

A lista de pacotes e dependências aparece primeiro:

- O nome do pacote *P* (dependente), sozinho, em uma linha. O nome terá no máximo 30 caracteres e serão apresentados no máximo 2500 pacotes.

- O nome de cada dependência de *P* nas próximas linhas, precedido por dois espaços e a palavra “*Depends:*”.

Para separar os pacotes e os comandos, será introduzida uma linha com três hífen. A partir desta separação, cada linha contém um comando:

- Os comandos podem ser “*install*” ou “*remove*”;
- Cada comando é seguido pelo nome de um dos pacotes especificados anteriormente.

Saída

1. Todos os pacotes instalados no sistema ao final da execução de todos os comandos, na ordem cronológica em que foram instalados (do mais antigo ao mais recente). No caso do exemplo anterior, a saída seria:

```
pkg_D  
pkg_C
```

Detalhes sobre comandos

1. Quando um pacote é instalado, todas suas dependências também são instaladas, recursivamente (ou seja, dependências de dependências).
 - a. Note que uma dependência deve ser sempre instalada **antes** do pacote dependente.
 - b. No entanto, podem existir ciclos na entrada, por exemplo: *A depende de B, B depende de C, C depende de A*. Para evitar *loops* infinitos, resolva da seguinte forma: **se um pacote é dependente de outro pacote que foi *marcado para instalação*, considere essa dependência como resolvida**. Neste exemplo, *C* seria instalado primeiro (sem exigir a instalação de *A*), depois *B*, depois *A*.
 - c. Se dois pacotes podem ser instalados em qualquer ordem – por exemplo, se *A depende de B* e *A depende de C*, *B* e *C* podem ser instalados em qualquer ordem – então deve ser respeitada a ordem em que os pacotes foram apresentados na entrada (ver *observações* abaixo). Neste caso, a ordem de instalação seria: *B, C, A*.
2. Se um pacote já estiver instalado e um comando de instalação for recebido, nada deve ser feito (ou seja, o *timestamp* da instalação não deve ser atualizado).
3. Quando um pacote é removido com “*remove*”, por padrão nenhuma de suas dependências é removida. No entanto, todos seus dependentes devem ser removidos, ou a estabilidade do sistema estaria comprometida.

Observações

1. A restrição da ordem de saída dos pacotes que podem ser instalados em qualquer ordem é necessária para que o sistema possa corrigir o problema sem ambiguidade. Para conseguir esse resultado, sempre que for realizar uma travessia

no grafo, ou seja, percorrendo as arestas de cada vértice, respeite a ordem em que os vértices foram apresentados na entrada. Se você utiliza uma implementação de grafo baseada em matriz de adjacências, esse resultado é natural; caso você use listas de adjacências, mantenha as arestas ordenadas pelo índice do vértice.

2. Cada caso de teste do sistema irá verificar uma pequena parte da implementação, começando com exemplos simples e continuando até os mais complexos. O comando *"install"*, por exemplo, será testado primeiro separadamente e depois em conjunto com *"remove"*. Se você conseguir implementar o exercício apenas parcialmente, submeta seu programa mesmo assim; a avaliação será em cima de cada caso de teste.
3. A entrada, os dados e os algoritmos utilizados neste exercício são todos baseados em dados reais de sistemas *Debian*, com poucas alterações. Ao resolver este exercício você estará, na prática, resolvendo um problema real de uma ferramenta utilizada diariamente por milhares de administradores de sistema no mundo.