

Projeto de Algoritmo com Implementação nº 0

MC458 — Projeto e Análise de Algoritmos I

Prof. Pedro J. de Rezende

2º Semestre de 2020

1 Introdução

Móbiles são geralmente encontrados pendurados sobre berços de bebês. A figura abaixo ilustra um móbile simples. Este consiste de uma haste, suspensa por um fio, com um objeto preso em cada lado. Um móbile também pode ser visto como uma espécie de alavanca, com um sustentáculo no ponto onde o fio sustenta a haste horizontal. Do princípio da alavanca, sabemos que, para equilibrar um móbile simples, os produtos do peso dos objetos pelas respectivas distâncias ao sustentáculo devem ser iguais. Isto é, $P_e D_e = P_d D_d$ onde P_e e D_e correspondem, respectivamente, ao peso e à distância do objeto da esquerda, e P_d e D_d ao peso e à distância do objeto da direita.

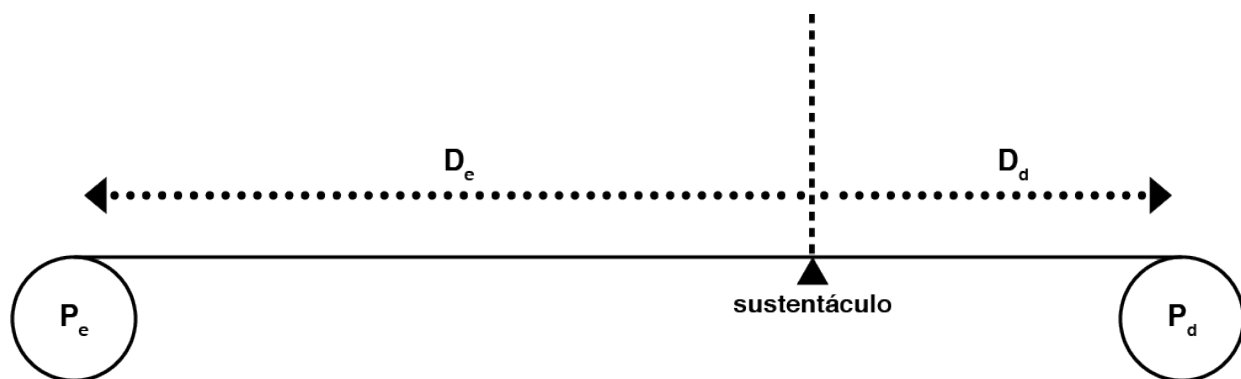


Figura 1: Móbile simples

Em um móbile mais complexo, cada objeto pode ser substituído por um sub-móbile, como ilustrado na próxima figura. Neste caso, não é tão simples verificar se um móbile está em equilíbrio e, para tanto, você deve projetar um algoritmo que, dada a descrição de um móbile, verifica se este está ou não em equilíbrio. Para que um móbile esteja em equilíbrio é necessário que todos os seus sub-móbiles estejam também em equilíbrio!

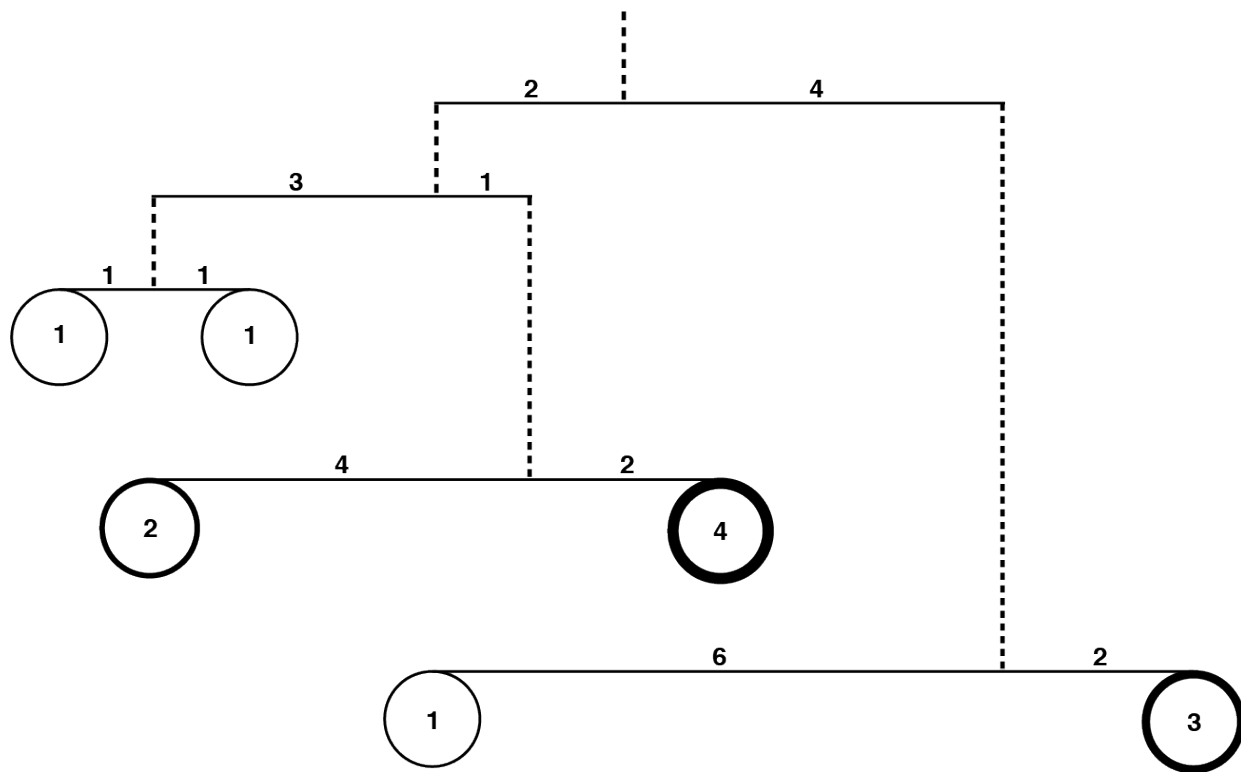


Figura 2: MóBILE com sub-móBILES

2 Objetivo

Para completar essa tarefa, você deverá seguir os passos aprendidos nesta disciplina para realizar o **Projeto de um Algoritmo** e sua **Implementação** para o problema abaixo. Isso inclui o projeto do algoritmo e sua prova de corretude (feitos de forma simultânea ou separadamente), sua análise de complexidade assintótica de pior caso, uma implementação e testes.

3 Problema

Dada a descrição de um móBILE, determinar se este está ou não em equilíbrio.

4 Entrada

Uma entrada é composta de diversas linhas. A primeira linha contém um único inteiro k indicando o número de linhas a seguir. Cada uma das k linhas seguintes é composta por

quatro inteiros separados por um espaço em branco. Estes quatro inteiros representam os pesos e as distâncias dos objetos ao sustentáculo, no formato $P_e D_e P_d D_d$.

Se P_e ou P_d for zero, então existe um sub-móvil pendurado na correspondente extremidade e as linhas seguintes definem este sub-móvil. Neste caso, computamos o peso do sub-móvil como sendo a soma dos pesos de todos os seus objetos, desconsiderando-se o peso das hastes e fios. Se ambos P_e e P_d tiverem valor zero, então as linhas seguintes definem dois sub-móveis: primeiro o esquerdo, e depois o direito. O exemplo abaixo representa o móvil desenhado acima.

```
5
0 2 0 4
0 3 0 1
1 1 1 1
2 4 4 2
1 6 3 2
```

O seu programa deve receber a entrada através da *entrada padrão*, i.e., dado um arquivo contendo a descrição de uma instância, você deve redirecionar o arquivo como entrada para o seu programa utilizando o operador <:

```
./mobile < instancia.in
```

5 Saída

A saída deve conter apenas um caractere: **S** se o móvil está em equilíbrio, ou **N** em caso contrário. Para a entrada acima, a saída correspondente seria:

S

6 Arquivos Providos

Para simplificar a implementação, as especificações citadas já foram implementadas em arquivos que podem servir como modelo para seu código. Baixe o código em [C](#) ou o código em [C++](#) de acordo com sua escolha de linguagem. O uso dos modelos não é obrigatório, porém, eles ajudam a evitar dificuldades com o formato de entrada e saída, já que discrepâncias em relação a esses formatos prejudicarão a avaliação da sua resolução.

7 Experimentos

Após a finalização do seu código, a implementação deve ser avaliada através de uma série de experimentos.

Antes de iniciá-los, lembre-se de compilar o código com a opção -O3 para ativar as rotinas de otimização do compilador! Utilize um dos seguintes comandos para compilação:

```
gcc -O3 mobile.c -o mobile
```

```
g++ -O3 mobile.cpp -o mobile
```

Para lhe ajudar na depuração do seu código, este [guia](#) descreve o uso básico dos debuggers *gdb* e *valgrind*.

8 Testes

Existe um conjunto de testes abertos (arquivos de entrada + saída) que está disponível para o uso durante a implementação. Além disso, existe um conjunto de testes fechados, aos quais você não terá acesso.

O programa será submetido a todos os testes dentro do Susy, porém o conjunto fechado não emitirá qual era a saída esperada, sendo apresentada apenas se a resposta estava correta ou incorreta.

Os testes abertos podem ser usados de forma automatizada via Susy, ou podem ser usados de maneira manual, baixando o arquivo testes.zip dos arquivos auxiliares, descompactando-o, e executando o script dentro da pasta testes, onde também deve ser colocado o seu código. O comando de execução deve ser:

```
./run_tests.sh
```

9 Critérios de avaliação

A avaliação consistirá na atribuição de conceitos para o Projeto:

- Inadequado se o Código não compila ou não passa nos testes abertos.
- Satisfatório se o Código compila, passa nos testes abertos e não passa nos testes fechados.
- Completo se o Código compila e passa em todos os testes.

A conversão desses conceitos para notas seguirá o que está descrito na [página da disciplina](#) levando-se ainda em conta a qualidade do conteúdo e completude do Relatório submetido.

10 Entrega

Observe os seguintes quesitos para realizar a entrega de seu trabalho:

- Prepare um arquivo pdf contendo um **Relatório** de **até duas páginas** descrevendo o Projeto do Algoritmo, sua prova de corretude e sua análise de complexidade e um arquivo contendo o **código** de sua implementação daquele algoritmo.
- A submissão do arquivo do Relatório, o qual deverá se chamar ‘PA#NNNNNN.pdf’ – onde # é o número do presente PA e NNNNNN é o seu R.A., deve ser feita via Google Classroom na Tarefa correspondente e esta Atividade.
- A sua submissão do arquivo de código será feita **exclusivamente** através do sistema SuSy. Para acessá-lo, basta utilizar seu R.A. e sua senha da DAC (Ex: **usuário:** 123456 e **senha:** senha-da-DAC).
- O arquivo com o seu código deve se chamar ‘mobile.c’ ou ‘mobile.cpp’, de acordo com a linguagem usada. O SuSy não irá alertá-lo sobre erros nos nomes, portanto, tenha atenção na hora de fazer suas submissões;
- No SuSy serão aceitas **até 5 submissões** por aluno, sendo preservada apenas a última submissão.

Link para submissão no SuSy: <https://susy.ic.unicamp.br:9999/mc458a/PA0>.

AVISO: as ferramentas de detecção de plágio do SuSy serão utilizadas durante a avaliação das submissões!