

Trabalho Prático 2: O problema da agenda de viagens de Rick Sanchez

Valor: 10 pontos

Data de entrega: 10 de novembro de 2019

Introdução

Está lembrado de Rick Sanchez? Ele ficou muito contente com a sua solução computacional para o problema das medições nos recipientes! Agora, ele precisa de você para resolver uma outra situação: como ele é um cientista de renome intergalático, recebe constantemente convites para visitar e dar palestras em diferentes planetas. Ele geralmente aceita o convite, desde que possa decidir o dia e o tempo que permanecerá no planeta, visitando-o. Além disso, Rick sempre aloca um tempo \mathbf{T} , em minutos, para visitaç o de planetas durante um m s.

Rick gostaria de automatizar a sua agenda de viagens interplanet rias. Para isso, ele pediu para voc  desenvolver um programa que, dado uma especifica  o de \mathbf{P} planetas e o tempo t que ele deve permanecer em cada um, informe como sa da uma agenda de visita  o dos planetas em cada m s, dado que o tempo total de visita  o em um m s n o ultrapasse o tempo \mathbf{T} .

Como todo cientista, Rick tem suas manias: ele quer visitar o m ximo de planetas poss veis no primeiro m s, ou seja, os k planetas de menor tempo de visita  o. Para o pr ximo m s, caso haja planetas ainda n o visitados, pretende novamente visitar o m ximo poss vel, considerando os $\{P - k\}$ planetas restantes, e assim sucessivamente. Al m disso, dados os k planetas, a visita  o deve acontecer seguindo a ordem alfab tica em k .

Rick ainda lhe prop e um grande desafio: ele garante que   poss vel organizar quais planetas devem ser visitados em cada m s com um algoritmo de complexidade de tempo $O(n \log_2 n)$. J  para indicar a ordem de visita  o dos planetas no m s, Rick garante que   poss vel fazer este processamento com complexidade de tempo $O(n \times k)$, dado que k   tamanho da cadeia de caracteres que identifica um planeta.

Detalhes do problema

O objetivo deste trabalho   praticar os conceitos relacionados a algoritmos cl ssicos de ordena  o. A seguir, s o esclarecidos detalhes relevantes que comp em o problema:

- \mathbf{T} indica o tempo em minutos alocados por Rick para visitar planetas durante 1 m s, tal que $T > 0$;
- S o especificados \mathbf{P} planetas, tal que $0 < P \leq 150000$. Cada especifica  o de um planeta $p \in P$ possui:
 - Um nome composto por uma cadeia de x caracteres (o tamanho da cadeia   o mesmo para todos os planetas), tal que $x < \log_2 P$;
 - Um tempo inteiro t em minutos, tal que $0 < t \leq T$;
- O primeiro m s de visita  o deve ter  ndice 1. Para cada m s de visita  o, o tempo total n o deve exceder \mathbf{T} ;

- Dados dois meses de visita  o subsequentes, M_1 e M_2 , o n  mero de planetas visitados em M_1 deve ser sempre maior ou igual o n  mero de planetas visitados em M_2 ;
- Para qualquer planeta p_1 agendado para o m  s M_1 , seu tempo t de visita  o deve ser sempre menor ou igual qualquer tempo de visita  o de um planeta p_2 de um m  s M_2 ;
- Para um determinado m  s M com k planetas agendados, o programa deve indicar para Rick que a visita  o deve ocorrer seguindo a ordem alfab  tica dos nomes dos k planetas.

Entrada e Sa  da

Os formatos desejados de entrada e sa  da s  o:

Entrada. Neste trabalho, a entrada ser   a padr  o do sistema (`stdin`). A primeira linha de uma inst  ncia de entrada do problema    composta por 3 valores inteiros T , P , x , correspondendo, respectivamente, ao tempo em minutos para visita  o de planetas durante 1 m  s, o n  mero total de planetas e o total de caracteres de cada nome de planeta. Ap  s a primeira linha, haver  o P linhas com informa   es sobre cada planeta. Cada uma das P linhas s  o compostas por um inteiro t e uma cadeia de caracteres de tamanho x , que representam, respectivamente, o tempo de visita  o do planeta e o seu nome.

Sa  da. Neste trabalho, a sa  da ser   a padr  o do sistema (`stdout`), e ela representar   a agenda de visita  o dos planetas. Ela deve ser composta por P linhas, sendo que cada uma deve possuir um inteiro M , uma cadeia de caracteres cp e um inteiro t , indicando, respectivamente, o m  s de visita  o, o planeta visitado e o tempo de visita  o daquele planeta. Note que, na sa  da, os meses devem ser listados de maneira ordenada. Al  m disso, para um determinado m  s M , os planetas devem aparecer de maneira alfabeticamente ordenada.

Exemplos

Exemplo de Entrada	Exemplo de Sa��da
720 5 2	1 aa 60
60 ag	1 ag 60
60 aa	1 dc 240
240 dc	1 zz 90
700 be	2 be 700
90 zz	

Exemplo de Entrada	Exemplo de Sa��da
300 9 3	1 aba 20
290 aaa	1 abb 20
10 xgf	1 dcz 100
100 dcz	1 fff 10
20 aba	1 xax 140
20 abb	1 xgf 10
10 fff	2 ggc 150
150 ggd	2 ggd 150
150 ggc	3 aaa 290
140 xax	

Entregáveis

Código-fonte. A implementação poderá ser feita utilizando as linguagens C ou C++. Não será permitido o uso da *Standard Library* do C++ ou de bibliotecas externas que implementem as estruturas de dados ou os algoritmos. **A implementação das estruturas e algoritmos utilizados neste trabalho deve ser sua. É expressamente proibido utilizar rotinas prontas de ordenação.** Os códigos devem ser executáveis em um computador com *Linux*. Caso não possua um computador com *Linux*, teste seu trabalho em um dos computadores do laboratório de graduação do CRC¹. A utilização de *Makefile*² é **obrigatória** para este trabalho.

Aplique boas práticas de programação e organize seu código-fonte em arquivos, classes e funções de acordo com o significado de cada parte. A separação de responsabilidades é um dos princípios da engenharia de software: cada função deve realizar apenas uma tarefa e cada classe deve conter apenas métodos condizentes com sua semântica.

Documentação. A documentação de seu programa **deverá** estar em formato **PDF**, **seguir** exclusivamente o modelo de trabalhos acadêmicos da SBC (que pode ser encontrado online³) e ser **sucinta**, não excedendo o limite de 7 páginas. Deverá conter **todos** os seguintes tópicos:

- Cabeçalho. Título do trabalho, nome e número de matrícula do autor.
- Introdução. Apresentação do problema abordado e visão geral sobre o funcionamento do programa.
- Implementação. Descrição sobre a implementação do programa. **Devem ser justificadas as escolhas dos algoritmos de ordenação implementados** (pode-se utilizar o material de aula da disciplina). Devem ser detalhados o funcionamento das principais funções e procedimentos utilizados, o formato de entrada e saída de dados, compilador utilizado, bem como decisões tomadas relativas aos casos e detalhes que porventura estejam omissos no enunciado. **Não devem ser inseridos trechos de código fonte na documentação.**
- Instruções de compilação e execução. Instruções de como compilar e executar o programa.
- Análise de complexidade. Estudo da complexidade de tempo e espaço do algoritmo de melhor e pior caso desenvolvido utilizando o formalismo da notação assintótica.
- Conclusão. Resumo do trabalho realizado, conclusões gerais sobre os resultados e eventuais dificuldades ou considerações sobre seu desenvolvimento.
- Bibliografia. Fontes consultadas para realização do trabalho.

O código-fonte e a documentação devem ser organizados como demonstrado pela árvore de diretórios na Figura 1. O diretório raiz deve ser nomeado de acordo seu **nome e último sobrenome**, separado por *underscore*, por exemplo, o trabalho de “Kristoff das Neves Björgman” seria entregue em um diretório chamado `kristoff_bjorgman`. Este diretório principal deverá conter um subdiretório chamado `src`, que por sua vez conterá os códigos (`.cpp`, `.c`, `.h`, `.hpp`) na estrutura de diretórios desejada. A documentação **em formato PDF** deverá ser incluída no diretório raiz do trabalho. Evite o uso de caracteres especiais, acentos e espaços na nomeação de arquivos e diretórios.

O diretório deverá ser submetido em um único arquivo ‘**nome_sobrenome.zip**’, (onde nome e sobrenome seguem as mesmas diretrizes para o nome do diretório, explicado acima) através do Moodle da disciplina até as **23:59** do dia **10 de novembro de 2019**.

¹<https://crc.dcc.ufmg.br/infraestrutura/laboratorios/linux>

²<https://opensource.com/article/18/8/what-how-makefile>

³<http://www.sbc.org.br/documentos-da-sbc/summary/169-templates-para-artigos-e-capitulos-de-livros/878-modelosparapublicaodeartigos>

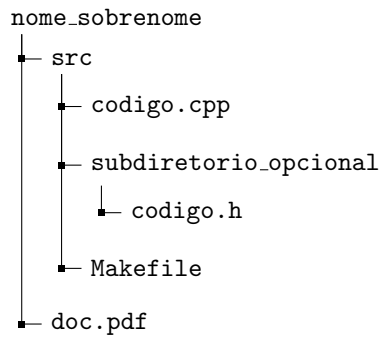


Figura 1: Estrutura de diretórios do entregável do TP2

Considerações Finais

Algumas considerações finais importantes:

- **Preste bastante atenção nos detalhes da especificação.** Cada detalhe ignorado acarretará em perda de pontos.
- O que será avaliado no trabalho:
 - Boas práticas de programação:** se o está código bem organizado e indentado, com comentários explicativos, possui variáveis com nomes intuitivos, modularizado, etc.
 - Implementação correta dos algoritmos:** se a árvore e o processo de decodificação foram implementados de forma correta e resolvem o problema aqui descrito.
 - Conteúdo da documentação:** se todo o conteúdo necessário está presente, reflete o que foi implementado e está escrito de forma coerente e coesa.
- Após submeter no Moodle seu arquivo ‘.zip’, faça o download dele e certifique-se que não está corrompido. Não será dada segunda chance de submissão para arquivos corrompidos.
- Em caso de dúvidas, **não hesite em perguntar** no Fórum de Discussão no Moodle ou procurar os monitores da disciplina – estamos aqui para ajudar!
- **PLÁGIO É CRIME:** caso utilize códigos disponíveis online ou em livros, **referencie** (inclua comentários no código fonte e descreva a situação na documentação). Trabalhos onde o plágio for identificado serão devidamente penalizados: o aluno terá seu trabalho anulado e as devidas providências administrativas serão tomadas. Discussões sobre o trabalho entre colegas são encorajadas, porém compartilhamento de código ou texto é plágio e as regras acima também se aplicam.
- Em caso de atraso na entrega, serão descontados $2^d - 1$ pontos, onde d é o número de dias (corridos) de atraso arredondado para cima.
- Comece o trabalho o mais cedo possível. Você nunca terá tanto tempo pra fazê-lo se começar agora!

Bom trabalho!

Apêndices

A Dicas para a documentação

O objetivo desta seção é apresentar algumas dicas para auxiliar na redação da documentação do trabalho prático.

1. **Sobre *Screenshots*:** ao incluir *screenshots* (imagens da tela) em sua documentação, evite utilizar o fundo escuro. Muitas pessoas preferem imprimir documentos para lê-los e imagens com fundo preto dificultam a impressão e visualização. Recomenda-se o uso de fundo branco com caracteres pretos, para *screenshots*.
2. **Sobre URLs e referências:** evite utilizar URLs da internet como referências. Geralmente URLs são incluídas como notas de rodapé. Para isto basta utilizar o comando `\footnote{\url{}}` no LaTeX, ou ativar a opção nota de rodapé⁴ no Google Docs/MS Word.
3. **Evite o Ctrl+C/Ctrl+V:** encoraja-se a modularização de código, porém a documentação é única e só serve para um trabalho prático. Reuso de documentação é auto-plágio⁵!

⁴<https://support.office.com/en-ie/article/insert-footnotes-and-endnotes-61f3fb1a-4717-414c-9a8f-015a5f3ff4cb>

⁵<https://blog.scielo.org/blog/2013/11/11/etica-editorial-e-o-problema-do-autoplagio/#.X0RgbdtKg5k>