

O objetivo deste trabalho consiste em implementar o método **secondMinimum()** que encontra o **segundo menor valor** em uma árvore binária de busca. Sua implementação não deve utilizar nenhum método auxiliar nem alterar a árvore. Caso não exista um segundo menor valor, uma mensagem de erro apropriada deve ser fornecida pelo método, porém não interrompa a execução utilizando **abort**.

Implementação

Você deve implementar o método conforme o enunciado fornecido anteriormente. Utilize apenas programação estruturada. Juntamente com este documento, estão sendo disponibilizados os arquivos (compactados em um arquivo ZIP):

- *BSTreeTemplate.h* (interface e implementação da classe *BynarySearchTree* usando *templates* em C++)
- *main.cpp* (programa principal para teste do método implementado)
- *StudentEmptyTest.h* (arquivo de teste do aluno, no formato da plataforma CxxTest¹)

Você pode incluir quaisquer subalgoritmos (funções, procedimentos ou métodos) que se fizerem necessários nestes arquivos, porém não remova ou altere os métodos já fornecidos (caso contrário, é possível que a plataforma CxxTest do Web-CAT atribua pontuação menor que o máximo admissível ao seu trabalho, mesmo que ele esteja correto).

Inclua, sob a forma de comentário no arquivo *BSTreeTemplate.h* o pior caso do método **secondMinimum()**, na notação assintótica **O(·)**. Indique claramente o que significa o parâmetro utilizado na notação assintótica **O(·)**.

Submissão

Submeta sua implementação no sistema Web-CAT, disponível em <http://kode.ffclrp.usp.br:8080/WebCat>. Este trabalho deve ser submetido individualmente.

Antes de submeter os arquivos necessários, coloque seu nome completo em todos os arquivos sendo submetidos, na forma de comentário no início de cada arquivo (.h ou .cpp).

Compacte os seguintes arquivos em um único arquivo .zip (não utilize espaços no nome do arquivo compactado, nem adicione pastas/diretórios no arquivo compactado):

- *BSTreeTemplate.h* (interface e implementação da classe *BynarySearchTree* usando *templates*)
- *StudentEmptyTest.h* (arquivo de teste do aluno, no formato da plataforma CxxTest)

Não inclua o programa principal, ou seja a função *main()*, na submissão ao Web-CAT. Respeite os nomes de arquivos, da classe e dos métodos. Submeta o arquivo compactado ao Web-CAT. Em caso de dúvida, procure o professor.

Avaliação

Na nota do trabalho, além dos critérios mencionados neste enunciado, também serão considerados os seguintes critérios (além daqueles já mencionados nas Disposições Gerais entregues no início do semestre):

- **Correção:** O programa faz o que foi solicitado? Faz tudo o que foi solicitado? Utiliza encapsulamento de informação? (i.e., acessa adequadamente os ADTs definidos?)

¹ Você pode inserir mais casos de teste neste arquivo, caso queira tenha interesse em testar com mais detalhes seu código. Consulte <http://cxxtest.com/guide.html#testAssertions>.

- **Eficiência:** As operações são executadas da maneira mais eficiente para cada estrutura de dados? Evita código duplicado/redundante/não atingível?
- **Interface:** É simples de usar, genérico, prático, tolera os erros mais óbvios? O trabalho foi entregue dentro das especificações (um arquivo *.h* para cada *.cpp* implementando um ADT? Os arquivos estão em formato ZIP, com os nomes de arquivos solicitados)?
- **Código fonte:** é claro (*layout*, espaçamento, organização em geral), nomes de variáveis são sugestivos, e há documentação/comentários apropriados no código? Faz uso de pré- e pós-condições? Quando aplicável, faz uso de subalgoritmos (funções, procedimentos ou métodos) adicionais que melhoram a legibilidade do(s) método(s) solicitado(s) sem comprometer sua eficiência (por exemplo, na notação assintótica $O(n)$, onde n representa o tamanho da entrada?)
- A análise do pior caso na notação assintótica $O(\cdot)$ foi incluída para o método **secondMinimum()**? Foi indicado claramente o tamanho da entrada?