OAT 2 - Ordenando e consultando os pedidos:

Oi, tudo bem?
Como estão as coisas?
Eu espero que esteja bem!

Eu tô sabendo da novidade. Parabéns, viu? Você foi super elogiado. O pessoal gostou muito do seu trabalho. Falaram até que esse seu trabalho vai ser colocado na amostra. A amostra vai ser dia 16/11, né? Eu vou estar lá, viu? Parabéns, mesmo! Mandou bem! Eu estou muito contente com a qualidade do seu trabalho.

Então, eu vim falar também das suas próximas atividades. Eu só trago notícias de trabalho, né? Eu prometo trazer notícias melhores na próxima. Quem sabe uma efetivação? (risos).

Uma vez que você já conhece um pouco do nosso motor de rotas, vamos falar do nosso sistema de indexação. O foodDelivery possui um conjunto enorme de pedidos que precisam ser geridos no menor tempo possível. Em geral, a nossa aplicação carrega um volume alto de pedidos na memória, de modo que inserir, excluir e buscar um item deve ser o mais prático e fácil possível. Por isso, os nossos pedidos estão indexados em relação ao par de valores (estabelecimento_id, pedido_id). O nosso indexador utiliza-se de uma implementação de uma estrutura de dados que garante complexidade assintótica $O(\log_2 n)$ em todas as operações de inserção, remoção e pesquisa. Nós fizemos uma implementação própria a fim de reduzir alguns gargalos que tínhamos. Contudo, em reunião com a equipe de engenharia, nós achamos interessante ver como vocês delis conseguiriam desenvolver um indexador simples, mas que já atendesse alguns requisitos que entendemos importantes para compreender a nossa implementação. Sendo assim, definimos os seguintes requisitos:

1. O seu indexador de usar como chave para indexação o par: estabelecimento_id e pedido_id.
2. O seu indexador deve disponibilizar as operações: inserir, excluir e buscar.
3. O seu indexador deve garantir uma complexidade $O(\log_2 n)$ nas operações de inserção, remoção e busca.

4. O seu indexador deve garantir que o índice cresça sempre balanceado e se mantenha balanceado a cada inserção ou remoção.

Bom, eu acho que é isso. O nosso objetivo com esse indexador é você compreender como muitas das buscas realizadas na nossa aplicação são feitas.

Produto:

Você deve implementar um indexador seguindo as restrições descritas por sua gestora. Além da implementação do indexador, você deve escrever um resumo técnico explicando as vantagens e limitações da sua solução. Algumas definições técnicas de projeto:

Linguagem de programação: Java, JavaScript ou Python.
Pontos de implementação:

1. **Solução técnica do indexador:** Implementação usando a linguagem de sua preferência entre as três estabelecidas.
2. **Visualizador de cada etapa da busca:** Durante a busca, é importante imprimir cada comparação realizada até encontrar o valor desejado. Você deve enumerar quantas comparações foram feitas no total.
3. **Visualizador do indexador:** Você deve implementar um modo de visualizar o modo como seu índice está estruturado. Pense como uma representação gráfica simples da sua estrutura. Não precisa ser nada sofisticado. Apenas devemos conseguir visualizar os dados indexados da mesma forma com que estão armazenados no indexador.

Resumo Técnico: Até duas laudas, explicando o contexto do problema e a solução adotada.

Curso: Sistemas de Informação
Disciplina: Complexidade de Algoritmos
Semestre: 2023.2
OAT 2

Pontuação:

Item	**Pontuação**
Indexador	12 pontos
Visualização de busca	2 pontos
Visualizador do indexador	4 pontos
Resumo Técnico	2 pontos
Total	**20 pontos**

Prazos:

O trabalho deve ser submetido no Blackboard até o dia **15/11/2023 às 06:00**.

Prazo máximo de atraso com moedas: 18/11/2023 às 6:00:
Custo por dia: 25 moedas.

Data e Hora	**Custo (total)**
16/11/2023 às 06:00	25 moedas
17/11/2023 às 06:00	50 moedas
18/11/2023 às 06:00	75 moedas