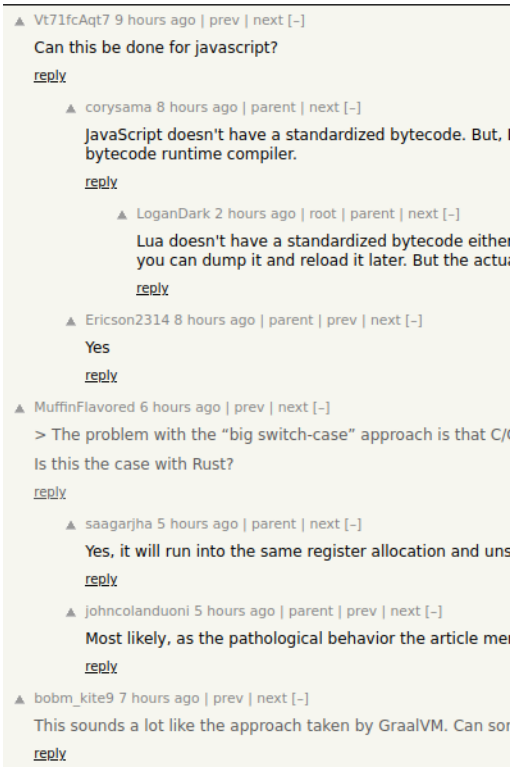
Tarefa (uso de GenAI permitido)

Você provavelmente já viu cadeias de comentários em uma página da web antes. Essas cadeias são usadas em sites de notícias e blogs como uma forma de facilitar discussões. Aqui está um exemplo retirado de uma: thread de comentários do Hacker News:



Em texto, esse tipo de cadeia poderia ser representado da seguinte forma:

```
# post 1
- comment 1.1
  - comment 1.1.1
  - comment 1.1.2
- comment 1.2

# post 2
- comment 2.1
- comment 2.2
```

Esses comentários podem ser aninhados em profundidade arbitrária, com muitas respostas a outras respostas.

Para este desafio, omitiremos o autor e os metadados geralmente vistos nesses comentários para focar no texto, IDs e relacionamentos entre eles.

Neste desafio, sua tarefa é escrever uma rota Express `GET /posts/42/comments` onde `42` pode ser qualquer identificador sequencial de post. Esta rota deve retornar uma estrutura JSON contendo todos os comentários associados a um post. A rota obterá os dados fazendo uma consulta ao pool `pg` pré-populado que se conecta a um banco de dados Postgres.

O esquema de tabela relevante que você usará para orientar suas consultas, junto com os dados que compõem os testes e a saída esperada, estão em `test/candidate.test.js`. Sua primeira tarefa é ler e entender o esquema do banco de dados neste arquivo e, em seguida, determinar a forma de resposta esperada lendo os casos de teste.

Na submissão, os esquemas permanecerão os mesmos, mas os dados serão alterados para garantir que seu código seja generalizado. Sua solução deve suportar quaisquer dados.

Recursos

Você pode consultar a documentação de JS, Node, SQL e pg enquanto desenvolve sua solução, além de ferramentas de AI.

Uma nota sobre timeouts da suite de testes

Os casos de teste podem dar erros de timeout sem mostrar erros se suas rotas forem assíncronas. Se seus testes derem erros de timeout, é provável que haja um erro de lógica em seu código impedindo que o manipulador de rota responda ao executor de teste. Tente adicionar um bloco try/catch e registre o erro para depurar seu código.