

Função `smallest_string_subsequences`

A função `smallest_string_subsequences` é responsável por determinar o tamanho da menor string que possui as strings `A` e `B` como subsequências. Ela utiliza o paradigma de programação dinâmica para resolver esse problema de maneira eficiente.

Parâmetros

- `x` (str): A primeira string de entrada, representada por `A`.
- `y` (str): A segunda string de entrada, representada por `B`.

Retorno

- `int`: O tamanho da menor string que possui as strings `A` e `B` como subsequências.

Funcionamento

A função utiliza uma matriz `dp` (tabela de programação dinâmica) para armazenar os resultados parciais. Ela itera pelas duas strings `A` e `B` e preenche a matriz `dp` com informações sobre o tamanho da menor string que contém `A` e `B` como subsequências.

- `size1` e `size2` representam o tamanho das strings `A` e `B`, respectivamente.
- A matriz `dp` é inicializada com zeros e tem dimensões $(size1 + 1) \times (size2 + 1)$ para armazenar os resultados parciais.
- O loop `for` aninhado itera por todas as posições da matriz `dp`.
- A matriz `dp[i][j]` representa o tamanho da menor string que contém os prefixos de tamanho `i` de `A` e `j` de `B` como subsequências.
- Se `i` é zero, significa que a string `A` está vazia, e o tamanho da menor string é igual ao tamanho da string `B` restante.
- Se `j` é zero, significa que a string `B` está vazia, e o tamanho da menor string é igual ao tamanho da string `A` restante.
- Se os caracteres em `A[i-1]` e `B[j-1]` forem iguais, então podemos estender a subsequência com ambos os caracteres, e o tamanho da menor string é o mesmo que para `A[i-1]` e `B[j-1]`.
- Caso contrário, precisamos escolher entre excluir o último caractere de `A` ou de `B`, e selecionamos a opção que resulta na menor string.