universität innsbruck

Advanced Functional Programming	WS 2025/2026	LVA 703139
Exercise Sheet 1, 10 points		Deadline: Tuesday, October 14, 2025, 4pm

- Solve the tasks in file `Exercise01.hs` and upload only this file in OLAT.
- Mark the solved exercises in OLAT.
- Your modified `Exercise01.hs` file must compile with `ghci` without error messages.

Task 1 *Optimal Brackets* **5 p.**

Design an algorithm `optBrackets :: [Integer] -> Brackets` that computes an optimal bracketing, represented by the following data type, where the integer in a split indicates the index of the matrix where the outermost brackets are added.

```
data Brackets = Leaf | Split Brackets Int Brackets
```

For instance, `Split (Split Leaf 0 Leaf) 1 (Split Leaf 2 (Split Leaf 3 Leaf))` represents the bracketing $(A_0 A_1)(A_2(A_3 A_4))$.
Your algorithm should be similar in structure to `optBracketCosts` from the lecture slides.

Task 2 *Embedding Relation* **5 p.**

First order terms are either variables or function symbols that are applied on lists of terms.
The following inference rules describe the embedding relation on terms.

- $\dfrac{s_1 \gtrsim_{emb} t_1 \quad \dots \quad s_n \gtrsim_{emb} t_n}{f(s_1, \dots, s_n) \gtrsim_{emb} f(t_1, \dots, t_n)}$ (args)
- $\dfrac{s_i \gtrsim_{emb} t}{f(s_1, \dots, s_n) \gtrsim_{emb} t}$ (sub)
- $\dfrac{}{x \gtrsim_{emb} x}$ (var)

For example, one can infer $f(m(x,y), s(z)) \gtrsim_{emb} f(y, s(z))$ and show $f(m(x,y), s(z)) \not\gtrsim_{emb} f(z, s(y))$.
In the template file you find an encoding of first order terms and a naive implementation of the embedding relation. It requires exponential time because of many overlapping recursive calls.
Design a more efficient Haskell function that decides $s \gtrsim_{emb} t$ for arbitrary input terms s and t. Your function should avoid overlapping recursive calls by using lazy dictionaries or lazy arrays.