BoT depends on the effectiveness of its thought generation structure. Thus, BoT utilizes the tree of thoughts (ToT) Yao et al. (2024), the most recent structure, as its base model to generate reasoning chains in each iteration. As mentioned in the main paper, the base thought generation model can also be the Graph of Thoughts (GoT) Besta et al. (2023), i.e., BoT with GoT. However, due to time constraints and the fact that the current GoT has not been applied to mathematical problems, BoT design exclusively embraces ToT. Furthermore, when employed as the base model within a boosting mechanism, thought structures in each iteration can remain lightweight. Ultimately, the proposed BoT generates heterogeneous tree structures, with each tree being a shallow-weighted binary tree.

C.1 Next thought generation and edge weights computation

Utilizing the *Prompt for the next thought generation* discussed in Section A, LLMs can generate the next possible thought for the reasoning step z_i by incorporating the experience $\mathbf{F}^{1}...t$ and replacing $\{G_i\}$ with $z_{1..,i-1}$. For a reasoning step z_i, LLMs utilize *Prompt for the thought evaluation* to generate the evaluation score as the edge weight between z_i and z_{i-1}. For a detailed procedure, the source code is available in *examples/BoostingOfThought/BoT_reasoner.py*. In a direct example of BoT applied to the '3 5 6 8' in the `Game of 24`, using gpt-3.5-turbo, Table 7 and Table 8 present the thought generation while the Table 9 show how to compute the computation.

C.2 The Necessity of Heterogeneous Tree Structures

In each iteration BoT, the heterogeneous tree structures with different tree growth strategies and various temperature and Top_p settings of LLMs are to be built to explore more reasoning search space and improve the robustness. As can be seen in the source code *BoostingOfThought/BoT_core.py* under the *examples/*, the temperature and Top_p values for LLMs within each tree are chosen from the ranges $[0.2, 0.4, 0.6, 0.7, 0.9, 1.1, 1.5]$ and $[0.1, 0.3, 0.5, 0.7, 0.9]$, respectively. And the tree growth strategy can either be level-wise growth or leaf-wise growth. We observed the following two benefits of guaranteeing such Heterogeneity.

Heterogeneity extends the reasoning search space, thus increasing the convergence speed. When different trees are constructed for distinct purposes, such as exploration with a level-wise strategy or exploitation with a leaf-wise strategy, and are based on LLMs with varying configurations for being random or deterministic, the generation of reasoning steps and the resulting reasoning chains can exhibit significant differences, effectively covering a wider range of reasoning possibilities. For example, in one iteration, when LLMs generate the next thought with more confidence, similar thoughts will be explored continuously; otherwise, LLMs with more randomness tend to generate diverse thoughts. It is generally challenging to predict whether deterministic reasoning or randomness can contribute to the solution. Therefore, incorporating heterogeneity by mixing different types and logical reasoning steps allows us to comprehensively explore the reasoning space within a single iteration, ultimately facilitating subsequent iterations. In the ablation study, we compare the performance of BoT between Heterogeneous and homogeneous tree structures.

Heterogeneity reduces the possibility of producing invalid or wrong reasoning chains, thus enhancing the robustness. Unlike heterogeneity, in trees with homogeneous settings, individual trees tend to generate thoughts following consistent logic and build reasoning chains with the same tree structures. Then, when the logic is wrong or the underlying structure is invalid for the current question, reasoning chains obtained by all trees of BoT in each iteration can only contain noisy and incorrect reasoning steps. Even after aggregating them to obtain a more refined reasoning chain for evaluation in BoT, the experience may still diverge significantly from providing suitable problem-solving suggestions. Therefore, designing tree thought structures to be heterogeneous can be a way to reduce the possibility that there are no effective reasoning chains to be evaluated for subsequent BoT's iteration. Therefore, designing tree thought structures to be heterogeneous can help mitigate the possibility of having no effective reasoning chains available for evaluation in subsequent BoT iterations. This enhancement of robustness allows BoT to tackle questions of varying difficulty levels.