构造具有单向性的正形置换

环境设置：

- Manjaro Linux x86_64, Kernel 4.19
- Python 3.8
- Numpy 1.18.5
- PyTorch 1.5.0 推荐 Cuda 加速版本，纯CPU版本也可运行

运行

- 最简单方法：运行`run.sh`

  ```
  bash run.sh
  ```

  cache 文件夹中的 full_permutations.txt 会存放着$q=0$时的128个构造方法。

- 手动筛选构造：

  1. 运行 `python 1_cache_expression_q0.py` 会筛选出$q=0$时每个布尔表达式的候选，筛选条件为是否平衡，最终结果储存在 cached_balance_expression.txt 中。如果想要筛选$q=1$的情况，可以运行 `python 1_cache_expression_q1.py`。
  2. 运行 `python 2_select_cached.py` 会组合 cached_balance_expression.txt 中的布尔表达式平衡候选，筛选出其中的置换，最终存放在 half_permutations.txt。
  3. 运行 `python 3_filter_selected.py` 会在 half_permutations.txt 中的每个置换上添加 n-1 次项和 n-2 次项，筛选出的置换会存放在 permutations.txt中。
  4. 运行 `python 4_split.py` 会自由组合 permutations.txt 中每个置换的前半部分和后半部分，筛选出其中的置换，并将最终结果储存在 full_permutations.txt 中。

- 观察逆向布尔函数：

  运行 `python inverse.py`，会打印出 permutations.txt 中第一个置换的逆向布尔函数，并统计每条布尔表示式有多少项。