**1. 功能概述**

reg\_search 函数用于从输入的文本中提取“标的证券”和“换股期限”信息。通过正则表达式，函数能够从给定的文本中识别出股票代码（六位数字+字母的组合）和日期（年份、月份、日期的格式），并将其以特定的格式输出。

**2. 输入**

* text (str)：待搜索的文本内容，通常是包含证券和换股期限信息的文档或字符串。

**3. 输出**

* formatted\_result (str)：格式化后的字符串，包含提取到的“标的证券”和“换股期限”的信息。
  + 标的证券：以列表形式返回，存储所有匹配的证券代码。
  + 换股期限：以列表形式返回，存储所有匹配的日期范围（格式化为 yyyy-mm-dd）。

**4. 正则表达式字典**

* **标的证券**：
  + **正则表达式**：r'(\d{6}\.\b[a-zA-Z]{2})'
  + **说明**：匹配股票代码，格式为六位数字，后跟一个点（.）和两个大写或小写字母。例如：600900.SH。
* **换股期限**：
  + **正则表达式**：r'(\d{4})\s\*年\s\*(\d{1,2})\s\*月\s\*(\d{1,2})\s\*日'
  + **说明**：匹配日期格式，如“2023 年 6 月 2 日”。通过三个分组分别匹配年份、月份和日期。

**5. 处理逻辑**

1. **标的证券**：
   * 使用正则表达式 (\d{6}\.\b[a-zA-Z]{2}) 匹配六位数字+点+两个字母的格式的股票代码，并将其存储在字典中。如果没有匹配项，则存储为 None。
2. **换股期限**：
   * 使用正则表达式 (\d{4})\s\*年\s\*(\d{1,2})\s\*月\s\*(\d{1,2})\s\*日 匹配类似“2023 年 6 月 2 日”的日期格式。匹配到的结果将会格式化为 yyyy-mm-dd 格式。例如：“2023 年 6 月 2 日”会被格式化为“2023-06-02”。如果没有匹配到，结果将为 None。

**6. 格式化输出**

* 输出的结果会是一个包含两项的字典：
  + '标的证券'：包含所有匹配到的证券代码，格式为列表。
  + '换股期限'：包含所有匹配到的换股期限，格式为列表，日期格式为 yyyy-mm-dd。
* 最终输出格式为：

[{

'标的证券': ['600900.SH'],

'换股期限': ['2023-06-02']

}]

**7. 示例输入与输出**

**输入：**

text = '''

标的证券：本期发行的证券为可交换为发行人所持中国长江电力股份

有限公司股票（股票代码：600900.SH，股票简称：长江电力）的可交换公司债

券。

换股期限：本期可交换公司债券换股期限自可交换公司债券发行结束

之日满 12 个月后的第一个交易日起至可交换债券到期日止，即 2023 年 6 月 2

日至 2027 年 6 月 1 日止。

'''

formatted\_result = reg\_search(text)

print(formatted\_result)

**输出：**

[{

'标的证券': ['600900.SH'],

'换股期限': ['2023-06-02']

}]

**8. 错误处理**

* 如果正则表达式没有匹配到任何数据，相关字段会返回 None。
* 如果未能找到预期的表格或文本格式，程序会打印相关的错误信息并提前退出。

**9. 注意事项**

* 本函数目前仅支持从中文格式的文本中提取“标的证券”和“换股期限”信息，对于其他格式的文本，可能需要根据需求进一步修改正则表达式或逻辑。
* 换股期限 的日期格式严格要求为“年 月 日”的格式，若格式不一致可能导致匹配失败。

**10. 性能**

由于正则表达式的匹配效率较高，本函数能够处理较长的文本数据，适合用于从较大的文档中提取证券信息。

**11. 适用场景**

* 提取金融或证券类文档中的相关信息（如债券发行、股权转换等）。
* 用于自动化处理证券类数据，如报告、公告等。