

```

/*
给你一个字符串 text，你需要使用 text 中的字母来拼凑尽可能多的单词 "balloon"（气球）。

字符串 text 中的每个字母最多只能被使用一次。请你返回最多可以拼凑出多少个单词 "balloon"。

示例 1:
输入: text = "nlaebolko"
输出: 1

示例 2:
输入: text = "loonbalxballpoon"
输出: 2

示例 3:
输入: text = "leetcode"
输出: 0

提示:

1 <= text.length <= 10^4
text 全部由小写英文字母组成

https://leetcode-cn.com/problems/maximum-number-of-balloons/
*/

```

分析:

- 统计需要的各种资源的数量,然后根据木桶原理返回最短的那块木板数量即可.

方法一:C++_木桶原理

```

class Solution
{
    // balloon
    // b * 1
    // a * 1
    // l * 2
    // o * 2
    // n * 1
public:
    int maxNumberOfBalloons(string text)
    {
        int count[5] = {0,};
        int i = 0;
        int minCount = 0;
        for(i = 0 ; i < text.size();i++)
        {
            switch(text[i])
            {
                case 'b':
                    count[0]++;

```

```

        break;
        case 'a':
            count[1]++;
        break;
        case 'l':
            count[2]++;
        break;
        case 'o':
            count[3]++;
        break;
        case 'n':
            count[4]++;
        break;
    }
}
count[2] /= 2;
count[3] /= 2;
minCount = count[0];
for(i=1;i<5;i++)
{
    if(count[i] < minCount)
    {
        minCount = count[i];
    }
}
return minCount;
}

};

/*
执行结果:
通过
显示详情
执行用时 :8 ms, 在所有 cpp 提交中击败了64.80%的用户
内存消耗 :8.6 MB, 在所有 cpp 提交中击败了100.00%的用户
*/

```