```
/*
给你一个有效的 IPv4 地址 address，返回这个 IP 地址的无效化版本。
所谓无效化 IP 地址，其实就是用 "[.]" 代替了每个 "."。
示例 1：
输入：address = "1.1.1.1"
输出："1[.]1[.]1[.]1"
示例 2：
输入：address = "255.100.50.0"
输出："255[.]100[.]50[.]0"

提示：
    给出的 address 是一个有效的 IPv4 地址
来源：力扣（LeetCode）
链接：https://leetcode-cn.com/problems/defanging-an-ip-address
著作权归领扣网络所有。商业转载请联系官方授权，非商业转载请注明出处。
*/
```

分析:

- 方法一：
  - 首先获取字符串的原始长度和点的数量，计算出返回需要的字符串长度，并申请对应长度空间;
  - 遍历原始字符串，若不为点就直接拷贝过去，若为点就换成对应字符串;
  - 注意字符串的结束标志要置0;
  - 时间复杂度$O(n)$

方法一:C

```
char* defangIPaddr(char* address)
{
    int     str_len     =   0       ;
    int     dot_count   =   0       ;
    int     i           =   0       ;
    int     j           =   0       ;
    char*   ret_val     =   NULL    ;

    while(address[i])
    {
        str_len++;
        if(address[i]=='.')
        {
            dot_count++;
        }
        i++;
    }

    ret_val = (char*)malloc(str_len+2*dot_count+1);
    ret_val[str_len+2*dot_count] = 0;
    i = 0;
    for(j=0;j<str_len;j++)
    {
```

```
        if(address[j]!='.')
        {
            ret_val[i++] = address[j];
        }
        else
        {
            ret_val[i++] = '[';
            ret_val[i++] = '.';
            ret_val[i++] = ']';
        }
    }
    return ret_val;
}
/*
执行结果:
通过
显示详情
执行用时 :0 ms, 在所有 C 提交中击败了100.00% 的用户
内存消耗 :6.8 MB, 在所有 C 提交中击败了100.00%的用户
*/
```

AlimyBreak
2019.09.01