125. Valid Palindrome (Easy)

Given a string `s`, return `true` if it is a palindrome, otherwise return `false`.

A palindrome is a string that reads the same forward and backward. It is also case-insensitive and ignores all non-alphanumeric characters.

Example 1:

```
Input: s = "Was it a car or a cat I saw?"

Output: true
```

Explanation: After considering only alphanumerical characters we have "wasitacaroracatisaw", which is a palindrome.

Example 2:

```
Input: s = "tab a cat"

Output: false
```

Explanation: "tabacat" is not a palindrome.

Constraints:

- `1 <= s.length <= 1000`
- `s` is made up of only printable ASCII characters.

▼ 思路:

作法一：two point比對

T:$O(n)$, S:$O(1)$

作法一

```
class Solution {
public:
    bool isPalindrome(string s) {
        int i=0,j=s.size()-1;
        while(j>i){
            s[i] = tolower(s[i]);
            s[j] = tolower(s[j]);
            if(!((s[i]<='z' && s[i]>='a') || (s[i]<='9'&& s[i]>='0'))){
                i++;
                continue;
            }
            if(!((s[j]<='z' && s[j]>='a') || (s[j]<='9'&& s[j]>='0'))){
                j--;
                continue;
            }
            if(s[i++]!=s[j--]) return false;
        }
        return true;
    }
};
```

作法一精簡

```
class Solution {
public:
    bool isPalindrome(string s) {
        int i=0,j=s.size()-1;
        while(j>i){
            s[i] = tolower(s[i]);
            s[j] = tolower(s[j]);
            if(!check_alphanum(s[i])){
                i++;
                continue;
            }
            if(!check_alphanum(s[j])){
                j--;
                continue;
            }
            if(s[i++]!=s[j--]) return false;
        }
        return true;
    }
    bool check_alphanum(char c){
        if((c<='z' && c>='a') || (c<='9'&& c>='0'))
            return true;
        else
            return false;
    }
};
```