1616 Split Two Strings to Make Palindrome

```
    bool isPalindrome(const string& s, int i, int j){
        while(i < j && s[i] == s[j]){
            i++;
            j--;
        }

        return i >= j;
    }

    bool check(const string& a, const string& b){
        int i = 0;
        int j = a.length() - 1;
        while(i < j && a[i] == b[j]){
            i++;
            j--;
        }

        return isPalindrome(a, i, j) || isPalindrome(b, i, j);
    }

    bool checkPalindromeFormation(string a, string b) {
        return check(a, b) || check(b, a);
    }
```

1642 Furthest Building You Can Reach

```
/*
    典型的贪心算法体现
```

```cpp

    因为 ladders 算作是无限的 bricks
    那么就先用 laaders 顶替， 然后 发现哪里bricks 用的少，就拿 bricks 补上
*/
class Solution {
public:
    int furthestBuilding(vector<int>& heights, int bricks, int ladders) {
        priority_queue<int, vector<int>, greater<int>> pq;

        for(int i = 0; i < heights.size() - 1; i++){
            int deltaHeight = heights[i + 1] - heights[i];
            if(deltaHeight > 0){
                pq.push(deltaHeight);
            }

            if(pq.size() > ladders){
                bricks -= pq.top();
                pq.pop();
            }

            if(bricks < 0)
                return i;
        }

        return heights.size() - 1;
    }
};
```