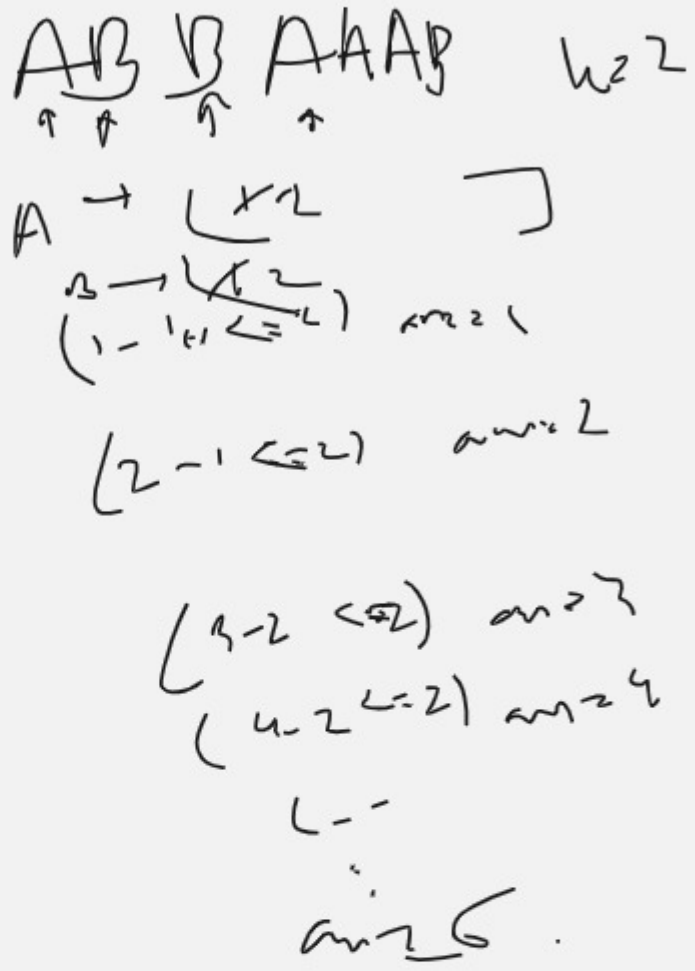
[424. Longest Repeating Character Replacement]

This I found actually quite tricky. Given in the problem, we need to find a maximum same character repeating substring and maximum of k character is allowed.

In the question it is given there will be be a string s and number of operations k allowed. We have to find what is the maximum length of same character substring that can form out of it.



The solution is as follows:

1. We take left and right pointers.
2. We keep count of frequency of the characters in a hashmap.
3. We increment the right pointer upto the end of the string and keep updating the count frequency and want to find which character is the most occuring in a window with a maxCnt.
4. If the number of non majority character is greater than k, we update left pointer++ and reduce the count frequency of character at left pointer.
5. In the loop, we keep track of largest sliding window we can obtain within k operations.
6. Return the answer.

```
class Solution {
public:
    int characterReplacement(string s, int k) {
        int ans = 0;
        int maxCnt = 0;
        int i = 0;
        unordered_map<char, int> mp;
        for(int j=0;j < s.size(); j++){
            mp[s[j]]++;
            maxCnt = max(maxCnt,mp[s[j]]);
            if(j -i + 1 - maxCnt > k) {
                mp[s[i]]--;
                i++;
            }
            ans = max(ans, j-i+1);
        }
        return ans;
    }
};
```