Data Structures and Algo in Java - Day 35

public class day35

{

public static void main (String [] args)

{

String s = "hehemalayalamhoho";

findLongestPalindromicSubstring(s);

String b = "aabcbaa";

findBeautyofString(b);

}

public static void findLongestPalindromicSubstring(String s)

{

if( s==null && s.length()<1)

{

return;

}

int start = 0;

int end = 0;

for(int i=0;i<s.length();i++)

{

int length1 = expandAroundCenter(s,i,i);

int length2= expandAroundCenter(s,i,i+1);

int length = Math.max(length1,length2);

if(length> end - start)

{

start = i-(length-1)/2;

end = i+length/2;

}

}

System.out.println(s.substring(start,end+1));

}

public static int expandAroundCenter(String s, int left, int right)

{

while(left>=0 && right<s.length() && s.charAt(left) == s.charAt(right))

{

left--;

right++;

}

return right - left -1;

}

public static void findBeautyofString(String s)

{

//1. find all substring

//1.5. find frequency of current substring

//2. find min and max freq

//3. find beauty

int n = s.length();

int beauty = 0;

for(int i=0;i<n;i++)

{

int [] freq = new int[26];

for(int j=i;j<n;j++)

{

char c = s.charAt(j);

freq[c-'a']++;

int maxFreq = 0;

int minFreq = Integer.MAX\_VALUE;

for(int k=0;k<26;k++)

{

if(freq[k]>0)

{

maxFreq = Math.max(maxFreq,freq[k]);

minFreq = Math.min(minFreq,freq[k]);

}

}

beauty = beauty + (maxFreq - minFreq);

}

}

System.out.println(beauty);

}

/\*

\* ✅ Step-by-Step Calculation for "aabcbaa"

Initialization:

• n = 7

• Total beauty = 0

⸻

i = 0

• Substring a → freq: a=1 → beauty: 0

• aa → a=2 → beauty: 0

• aab → a=2, b=1 → 2 - 1 = 1

• aabc → a=2, b=1, c=1 → 2 - 1 = 1

• aabcb → a=2, b=2, c=1 → 2 - 1 = 1

• aabcba → a=3, b=2, c=1 → 3 - 1 = 2

• aabcbaa → a=4, b=2, c=1 → 4 - 1 = 3

Sum (i = 0): 0 + 0 + 1 + 1 + 1 + 2 + 3 = 8

⸻

i = 1

• a → a=1 → 0

• ab → a=1, b=1 → 1 - 1 = 0

• abc → a=1, b=1, c=1 → 1 - 1 = 0

• abcb → a=1, b=2, c=1 → 2 - 1 = 1

• abcba → a=2, b=2, c=1 → 2 - 1 = 1

• abcbaa → a=3, b=2, c=1 → 3 - 1 = 2

Sum (i = 1): 0 + 0 + 0 + 1 + 1 + 2 = 4

⸻

i = 2

• b → b=1 → 0

• bc → b=1, c=1 → 0

• bcb → b=2, c=1 → 2 - 1 = 1

• bcba → b=2, c=1, a=1 → 2 - 1 = 1

• bcb-aa → b=2, c=1, a=2 → 2 - 1 = 1

Sum (i = 2): 0 + 0 + 1 + 1 + 1 = 3

⸻

i = 3

• c → c=1 → 0

• cb → c=1, b=1 → 0

• cba → c=1, b=1, a=1 → 0

• cbaa → c=1, b=1, a=2 → 2 - 1 = 1

Sum (i = 3): 0 + 0 + 0 + 1 = 1

⸻

i = 4

• b → b=1 → 0

• ba → b=1, a=1 → 0

• baa → b=1, a=2 → 2 - 1 = 1

Sum (i = 4): 0 + 0 + 1 = 1

⸻

i = 5

• a → a=1 → 0

• aa → a=2 → 0

Sum (i = 5): 0 + 0 = 0

⸻

i = 6

• a → a=1 → 0

Sum (i = 6): 0

\*/

}