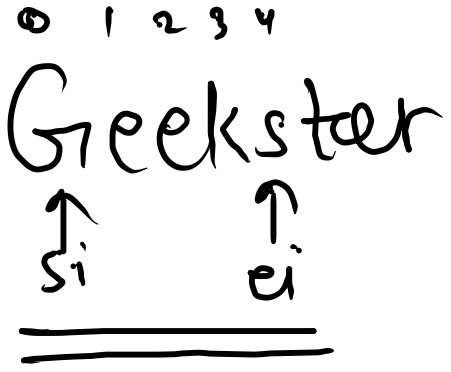
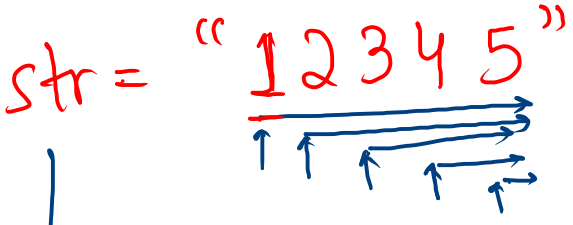
Substring (inbuilt)



1) str.substring(si, ei+1)
(0, 5)
→ // Geeks

2) str.substring(si); // Geekster

Sum of All Substrings



↳ substrings

1

1 2

1 2 3

1 2 3 4

1 2 3 4 5

2

2 3

2 3 4

2 3 4 5

3

3 4

3 4 5

4

4 5

5

Code

```java
public static void main(String[] args) {
    Scanner scn = new Scanner(System.in);
    String str = scn.nextLine();

    System.out.println(sumOfSubstrings(str));
}

public static int sumOfSubstrings(String str) {
    int sum = 0;
    for (int i = 0; i < str.length(); i++) {
        for (int j = i; j < str.length(); j++) {
            sum += Integer.valueOf(str.substring(i, j + 1));
        }
    }
    return sum;
}
```

Inbuilt functions

- Integer.parseInt(str);
- Integer.valueOf(str);

Desired String

str = "ABADA"

↳ substring

ABA
ADA
ABADA

j = i+1

AB
ABA
ABAD
ABADA

BA
BAD
BADA
AD
ADA , DA

Code

```java
public static void desiredString(String str) {
    int count = 0;
    int len = 0;
    String longestSub = "";
    for (int i = 0; i < str.length(); i++) {
        for (int j = i + 1; j < str.length(); j++) {    // single char case will exclude
            String sub = str.substring(i, j + 1);
            if ( sub.charAt(0) == 'A' && sub.charAt(sub.length() - 1) == 'A' ) {
                count++;
                if ( len < sub.length() ) {
                    len = sub.length();
                    longestSub = sub;
                }
            }
        }
    }

    if (count != 0) {
        System.out.println(count);
        System.out.println(len);
        System.out.println(longestSub);
    } else {
        System.out.println(-1);
    }
}
```

ls = ~~""~~ "~~ABA~~" "ABADA"
c = ~~0~~ ~~1~~ ~~2~~ 3
l = ~~0~~ ~~3~~ 5

str = "ABADA"

AB
ABA len = 3
ABAD
ABADA len = 5
BA
BAD
BADA
AD
ADA len = 3
DA

Power of a String (find substring of max. length with a unique char)

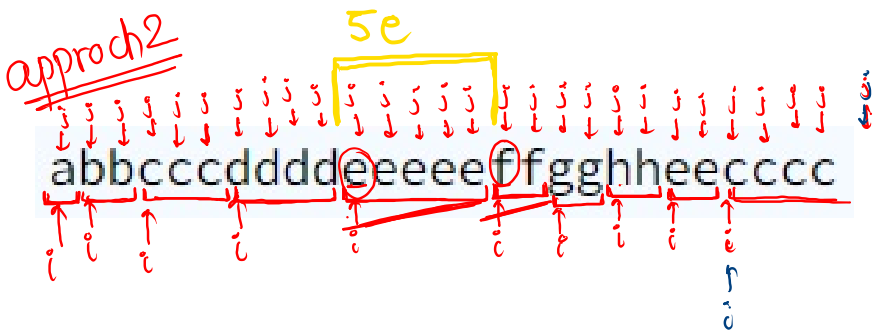
abbcccddddeeeeeffgghheecccc

- a
- bb
- ccc
- dddd
- eeeee ← 5
- ff
- gg
- hh
- ee
- cccc

approch 1

- generate all substrings
- if char are unique or not (array as hashmap) (count same char)
- keep updating for better ans

approch2



Count length = 1 2 3 4 5

T.C = O(N)

N → len of str

{ ans = 0
  count = 1

count = 1 1 2 1 2 3 1 2 3 4 1 2 3 4 5 1 2 1 2 1 2 1 2 1 2 3 4

ans = 0 1 2 3 4 5

```
public static int powerString(String str) {
    int i = 0;
    int count = 1;
    int ans = 0;
    while (i < str.length()) {
        int j = i + 1;
        while (j < str.length()) {
            if ( str.charAt(i) != str.charAt(j) ) {
                ans = Math.max( ans, count );
                count = 1;
                break;
            }
            count++;
            j++;
        }
        i = j;
    }
    ans = Math.max( ans, count );
    return ans;
}
```

factory

i=0, j=1

i=1, j=2 3

i=3, j=4 5 6

i=6, j=7 8 9 10

i=10, j=11 12 13 14 15

i=15, j=16 17

i=17, j=18 19

i=19, j=20 21

i=21, j=22 23

i=23, j=24 25 26 27