

سید علی نولیند - ۹۸۲۰۷۱۳ - پروژه طراحی الگوریتم ۲ - اسناد احمد



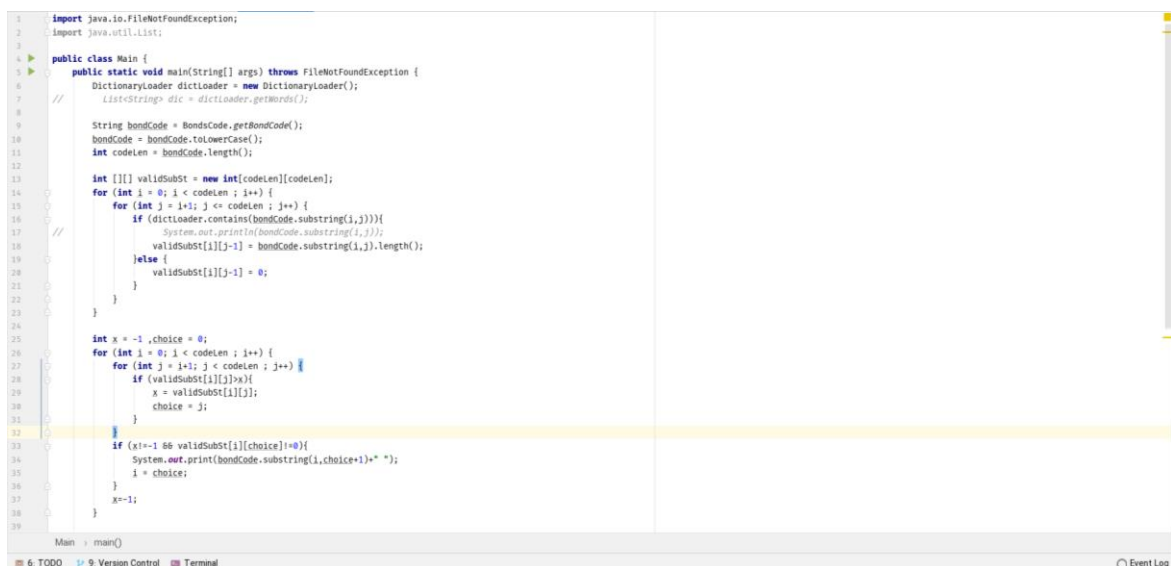
```
Run: Main x
"D:\IntelliJ IDEA 2019.3.3\install\jbr\bin\java.exe" "-javaagent:D:\IntelliJ IDEA 2019.3.3\install\lib\idea_rt.jar=50983:D:\IntelliJ IDEA 2019.3.3\install\bin" -Dfile.encoding=UTF-8
Bond code to MI6:
ABORTTHEPLANMEETATTHEDARKCABIN
abort the plan meet at the dark cabin
Process finished with exit code 0
```

شکل ۱



```
Run: Main x
"D:\IntelliJ IDEA 2019.3.3\install\jbr\bin\java.exe" "-javaagent:D:\IntelliJ IDEA 2019.3.3\install\lib\idea_rt.jar=50854:D:\IntelliJ IDEA 2019.3.3\install\bin" -Dfile.encoding=UTF-8
Bond code to MI6:
CALLSECURITYATMIAIRPORTBECAUSEITHINKABOMBISABOUTTOGOOFF
call security at airport because it hi bomb about to go off
Process finished with exit code 0
```

شکل ۲



```
1 import java.io.FileNotFoundException;
2 import java.util.List;
3
4 public class Main {
5     public static void main(String[] args) throws FileNotFoundException {
6         DictionaryLoader dictLoader = new DictionaryLoader();
7         List<String> dic = dictLoader.getWords();
8
9         String bondCode = BondsCode.getBondCode();
10        bondCode = bondCode.toLowerCase();
11        int codeLen = bondCode.length();
12
13        int [][] validSubst = new int[codeLen][codeLen];
14        for (int i = 0; i < codeLen; i++) {
15            for (int j = 1; j <= codeLen; j++) {
16                if (dictLoader.contains(bondCode.substring(i, j))) {
17                    System.out.println(bondCode.substring(i, j));
18                    validSubst[i][j-1] = bondCode.substring(i, j).length();
19                } else {
20                    validSubst[i][j-1] = 0;
21                }
22            }
23        }
24
25        int x = -1, choice = 0;
26        for (int i = 0; i < codeLen; i++) {
27            for (int j = 1; j <= codeLen; j++) {
28                if (validSubst[i][j] > x) {
29                    x = validSubst[i][j];
30                    choice = j;
31                }
32            }
33            if (x != -1 && validSubst[i][choice] != 0) {
34                System.out.print(bondCode.substring(i, choice+1) + " ");
35                i = choice;
36            }
37            x = -1;
38        }
39    }
40}
```

شکل ۳

برنامه به این صورت کار میکند که ابتدا همه ی زیر رشته های ممکن در کد باندرابررسی میکند و اگر معنا دار باشد طول کلمه معنا دار را در جایگاه مربوط به خودش در آرایه valiSubSt قرار میگیرد و در غیر این صورت .

سپس به واسطه یک حلقه ی تو در تو دیگر همه ی این خانه ها بررسی میشوند و از هر ستون (مساوی) بلندترین زیررشته انتخاب شده و چاپ می شود. که البته این الگوریتم فقط در زمان هایی کاملاً درست کار میکند که در میانه جمله کلمه تک حرفی که میتواند با حرف (های) بعضی اش کلمه درست کند وجود نداشته باشد

از نظر پیچیدگی زمان قسمت اول در صورتی که طول رشته کد را n در نظر بگیریم $n^2/2$ است. و قسمت دوم در بهترین حالت $n^2/2$ است که در n های کم رخ می دهد چون کلمات از حی بیشتر کاراکتر ندارند. پس میتوان گفت پیچیدگی زمان این پروژه $O(n^2)$ است.