all indices

arr : [2, 9, 8, 2, 9, 6, 4, 8, 9, 7, 8]
0 1 2 3 4 5 6 7 8 9 10

[1, 4, 8]

ele = 9

jsj 0 0 1 1 1 2 2 2 2 3 3

arr : [2, 9, 8, 2, 9, 6, 4, 8, 9, 7, 8]
0 1 2 3 4 5 6 7 8 9 10

↑

1	4	8
0	1	2

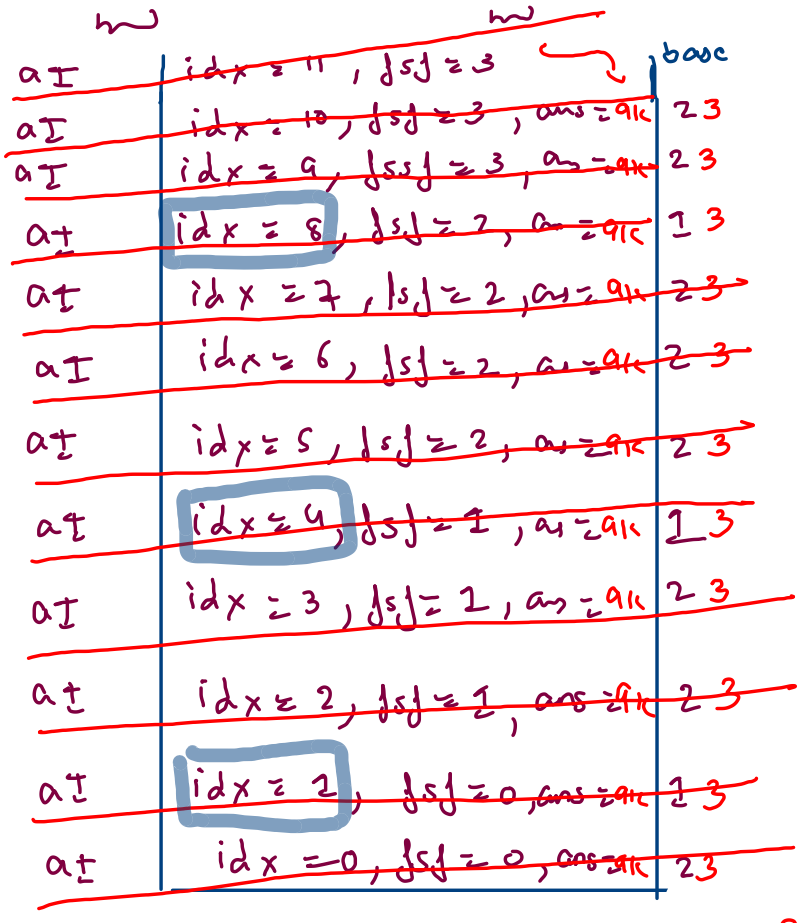
ans[jsj] = idx

arr : [2, 9, 8, 2, 9, 6, 4, 8, 9, 7, 8]

0 1 2 3 4 5 6 7 8 9 10

```
public static int[] allIndices(int[] arr, int x, int idx, int fsf) {
    if(idx == arr.length) {
        return new int[fsf];
    }

    int[]ans;

    if(arr[idx] == x) {
        ans = allIndices(arr,x,idx+1,fsf+1);
        ans[fsf] = idx;
    }
    else {
        ans = allIndices(arr,x,idx+1,fsf);
    }

    return ans;
}
```

1
2
3



arr = 4ik

X = 9

1	4	8
0	1	2

9ik

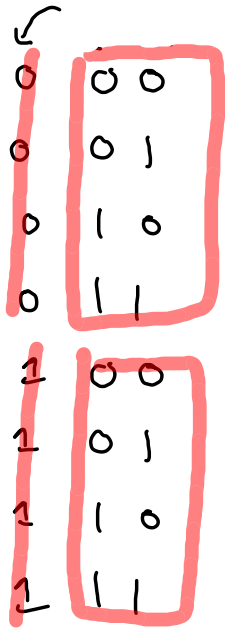
get subseq

a b c n = 3

0 0 0	- - -	⟶ .
0 0 1	- - c	⟶ c
0 1 0	- b -	⟶ b
0 1 1	- b c	⟶ b c
1 0 0	a - -	⟶ a
1 0 1	a - c	⟶ a c
1 1 0	a b -	⟶ ab
1 1 1	a b c	⟶ abc

[. , c, b, bc, a, ac, ab, abc]

a b c



v2

[·, c]

[·, c, b, bc]

[·, c, b, bc, → a X

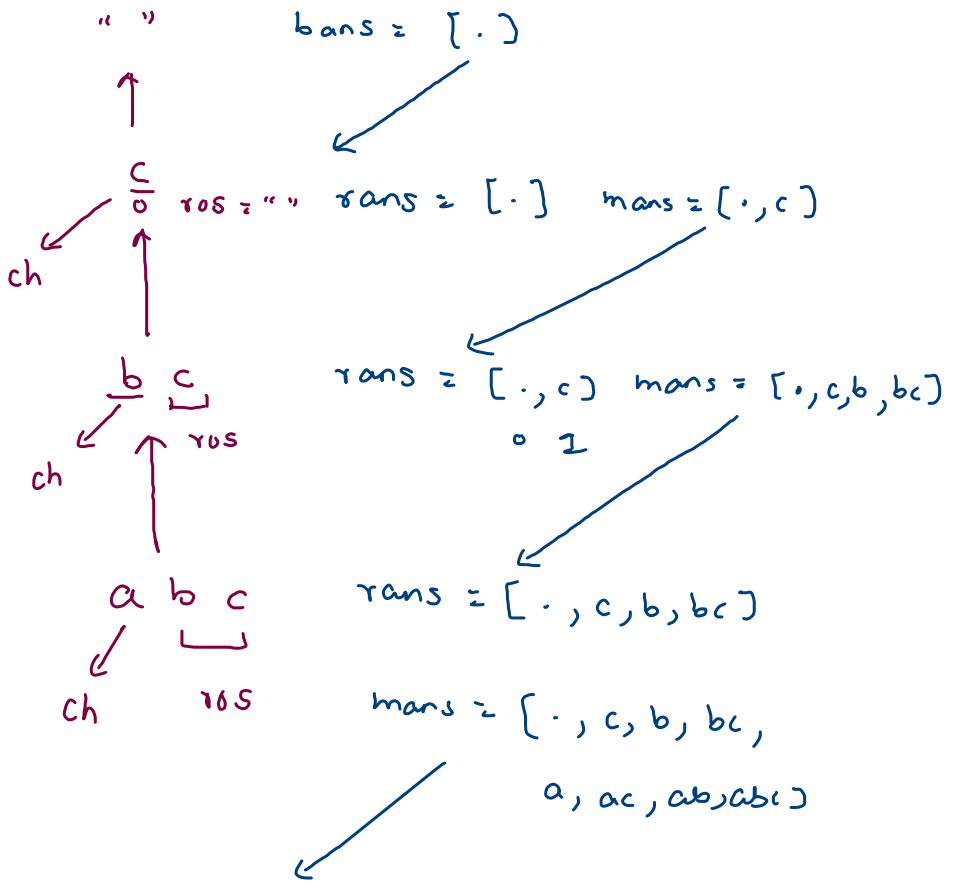
a, ac, ab, abc] → a ∪

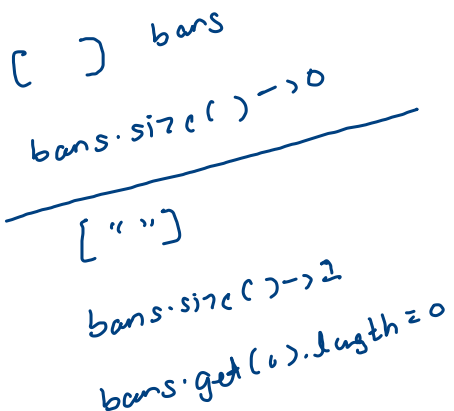
```java
public static ArrayList<String> gss(String str) {
    if(str.length() == 0) {
        ArrayList<String>bans = new ArrayList<>();
        bans.add("");
        return bans;
    }

    char ch = str.charAt(0);
    String ros = str.substring(1);

    ArrayList<String>rans = gss(ros); //faith
    ArrayList<String>myans = new ArrayList<>();


    //ch -> no choice
    for(int i=0; i < rans.size();i++) {
        myans.add(rans.get(i));
    }

    //ch -> yes choice
    for(int i=0; i < rans.size();i++) {
        myans.add(ch + rans.get(i));
    }

    return myans;
```

pre

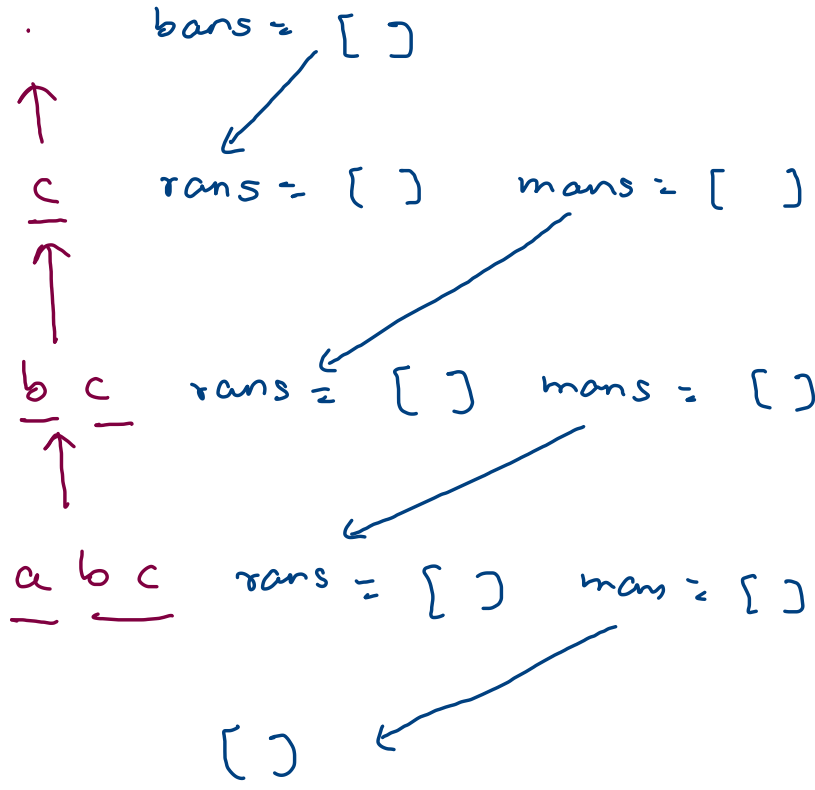
call

post





```java
public static ArrayList<String> gss(String str) {
    if(str.length() == 0) {
        ArrayList<String>bans = new ArrayList<>();
        bans.add("");
        return bans;
    }

    char ch = str.charAt(0);
    String ros = str.substring(1);

    ArrayList<String>rans = gss(ros); //faith
    ArrayList<String>myans = new ArrayList<>();


    //ch -> no choice
    for(int i=0; i < rans.size();i++) {
        myans.add(rans.get(i));
    }

    //ch -> yes choice
    for(int i=0; i < rans.size();i++) {
        myans.add(ch + rans.get(i));
    }

    return myans;
```



0 -> .;
1 -> abc
2 -> def
3 -> ghi
4 -> jkl
5 -> mno
6 -> pqrs
7 -> tu
8 -> vwx
9 -> yz

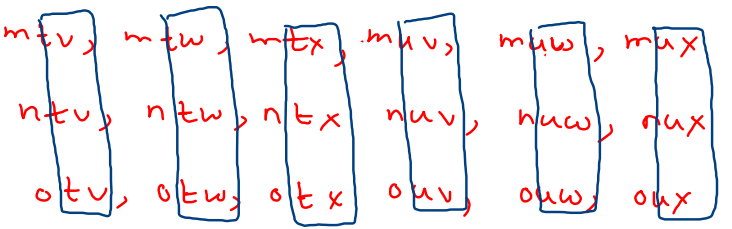
str : " 5 7 8 "

5 7 8

m t v

n u w

o x



[tv, tw, tx, uv, uw, ux)

"78"

3 x 2 x 3 = 18

78 : [tv, tw, tx, uv, uw, ux]

578 : [mtv, mtw, mtx, muv, muw, mux, 'm'

ntv, ntw, ntx, nuv, nuw, nux, 'n'

otv, otw, otx, ouv, ouw, oux) 'o'

Ascii → american standard code for information inter-change

① a-z → 97-122

- a → 97
- b → 98
- ⋮
- z → 122

② A-Z → 65-90

- A → 65
- B → 66
- C → 67
- ⋮
- Z → 90

③ '0'-'9' → 48-57

- '0' → 48
- '1' → 49
- '2' → 50
- ⋮
- '9' → 57

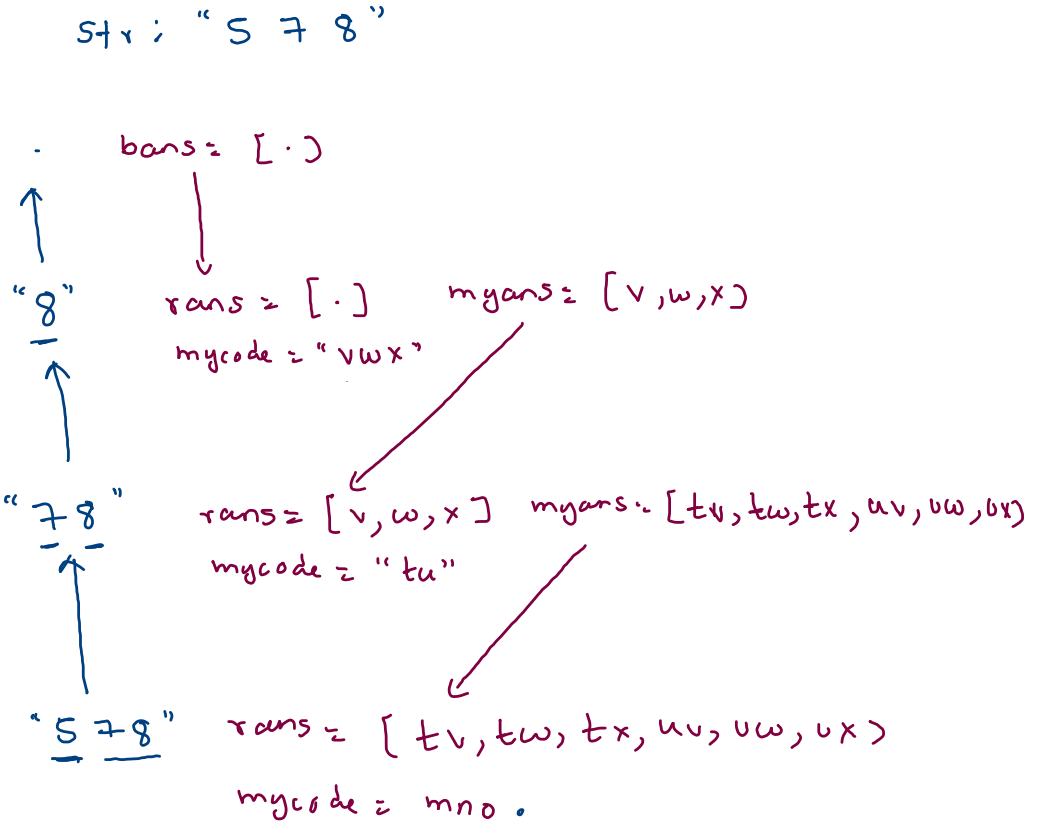
```
str = "s78"
char ch = str[0]
    ch = 's'
    codes [ch]
    codes ['s']
    = codes [83]
```

'3' = '0' + 3

'3' - '0' = 3

0 -> .;
1 -> abc
2 -> def
3 -> ghi
4 -> jkl
5 -> mno
6 -> pqrs
7 -> tu
8 -> vwx
9 -> yz

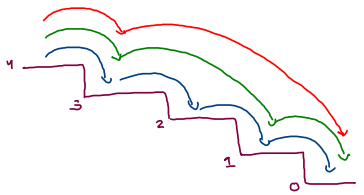
```java
public static ArrayList<String> getKPC(String str) {
    if(str.length() == 0) {
        ArrayList<String>bans = new ArrayList<>();
        bans.add("");
        return bans;
    }

    char ch = str.charAt(0);
    String ros = str.substring(1);

    ArrayList<String>rans = getKPC(ros);
    ArrayList<String>myans = new ArrayList<>();

    String mycode = code[ch-'0'];

    for(int i=0; i < mycode.length();i++) {
        char mch = mycode.charAt(i);

        for(int j=0; j < rans.size();j++) {
            myans.add(mch + rans.get(j));
        }
    }

    return myans;
```



myans = [mtv, mtw, mtx, muv, muw, mux,
ntv, ntw, ntx, nuv, nuw, nux,
otv, otw, otx, ouv, ouw, oux]

Stair path

n = 4



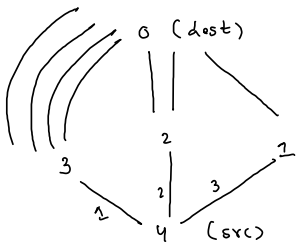
1 jump
2 jump
3 jump

4 to 0 ways : [1111, 121, 13, 211, 112, 31, 22]

3 to 0 ways : [111, 12, 21, 3]

4 to 0 ways : 4 to 3 + 3 to 0 way

1111, 112, 121, 13 ✗ (we did not get all ans)



4 to 0 = 4 to 3 + 3 to 0

'1' [111, 12, 21, 3]

4 to 2 + 2 to 0

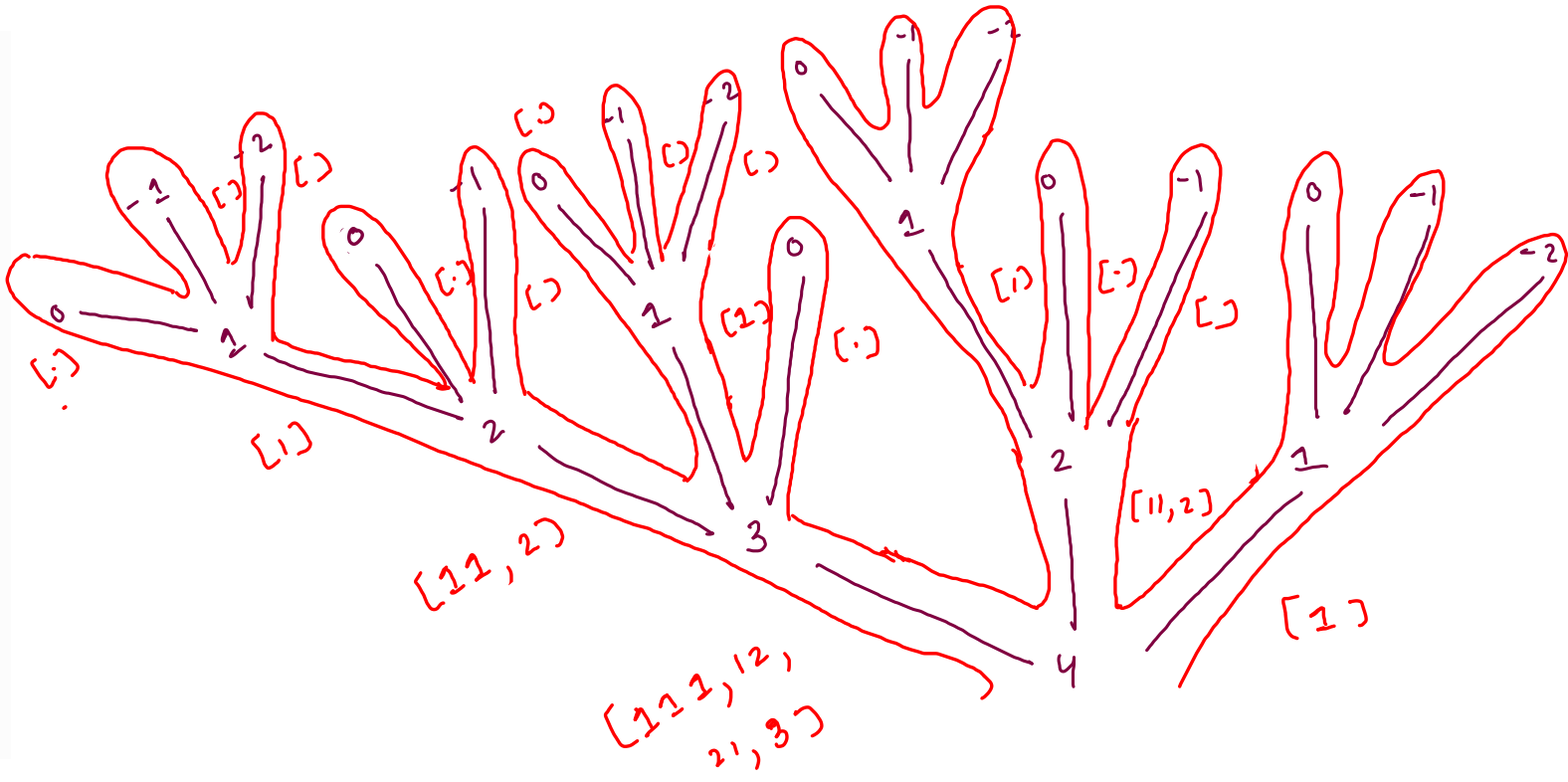
'2' [11, 2]

4 to 1 + 1 to 0

'3' [1]

ways(n to 0) = ways(n-1 to 0) + ways(n-2 to 0) + ways(n-3 to 0)

```java
public static ArrayList<String> getStairPaths(int n) {
    if(n == 0) {
        ArrayList<String>bans = new ArrayList<String>();
        bans.add("");
        return bans;
    }

    if(n < 0) {
        ArrayList<String>bans = new ArrayList<String>();
        return bans;
    }

    ArrayList<String>nm1to0 = getStairPaths(n-1); //n-1 to 0
    ArrayList<String>nm2to0 = getStairPaths(n-2); //n-2 to 0
    ArrayList<String>nm3to0 = getStairPaths(n-3); //n-3 to 0

    ArrayList<String>nto0 = new ArrayList<>();

    //n to 0 ->  '1' + n-1 to 0
    for(int i=0; i < nm1to0.size();i++) {
        nto0.add('1' + nm1to0.get(i));
    }

    //n to 0 -> '2' + n-2 to 0
    for(int i=0; i < nm2to0.size();i++) {
        nto0.add('2' + nm2to0.get(i));
    }

    //n to 0 -> '3' + n-3 to 0
    for(int i=0; i < nm3to0.size();i++) {
        nto0.add('3' + nm3to0.get(i));
    }

    return nto0;

}
```



```java
public static ArrayList<String> getStairPaths(int n) {
    if(n == 0) {
        ArrayList<String>bans = new ArrayList<String>();
        bans.add("");
        return bans;
    }

    if(n < 0) {
        ArrayList<String>bans = new ArrayList<String>();
        return bans;
    }

    ArrayList<String>nm1to0 = getStairPaths(n-1); //n-1 to 0
    ArrayList<String>nm2to0 = getStairPaths(n-2); //n-2 to 0
    ArrayList<String>nm3to0 = getStairPaths(n-3); //n-3 to 0

    ArrayList<String>nto0 = new ArrayList<>();

    //n to 0 ->  '1' + n-1 to 0
    for(int i=0; i < nm1to0.size();i++) {
        nto0.add('1' + nm1to0.get(i));
    }

    //n to 0 -> '2' + n-2 to 0
    for(int i=0; i < nm2to0.size();i++) {
        nto0.add('2' + nm2to0.get(i));
    }

    //n to 0 -> '3' + n-3 to 0
    for(int i=0; i < nm3to0.size();i++) {
        nto0.add('3' + nm3to0.get(i));
    }

    return nto0;
}
```