

Code@Feit

Квадратен корен (2)

```
public static int sqrt (int n){  
    koren (n, 1, n);  
}
```

```

public static int koren(int n, int down, int up){
    if (down*down == n) {
        return down;
    } else if (up*up == n) {
        return up;
    } else if ((up-down)==1){
        return down;
    } else {
        int mid=(up+down)/2;
        if (mid*mid==n) {
            return mid;
        } else if (mid*mid > n) {
            return koren(n, down, mid);
        } else {
            return koren(n, mid, up);
        }
    }
}

```

```

public static int koren(int n){
    int down = 1;
    int up = n;
    if (down*down == n) return down;
    if (up*up == n) return up;
    while (up-down !=1){
        int mid=(up+down)/2;
        if (mid*mid==n) {
            return mid;
        } else if (mid*mid > n) {
            up=mid;
        } else {
            down=mid;
        }
    }
    return down;
}

```

Анализа

- Простор
- Runtime

- Квадратен корен (1), (2)
- Фибоначи (1), (2)

- Броење
- Лимити
- Графови

Квадратен корен (1)

```
public static int koren(int n, int i){
    if (i*i==n) {
        return i;
    } else {
        return koren(n, i+1);
    }
}
```

Фибоначи (1)

```
public static int recFib(int n){
    if (n==0 || n==1) {
        return n;
    }
    else {
        return recFib(n-1)+recFib(n-2);
    }
}
```

Факторијал + Сигма

```
public static int f&s(int n){
    System.out.println(n+"="+faktorijal(n));
    if (n==1){
        return 1;
    } else {
        return n+f&s(n-1);
    }
}
```

Фибоначи (2)

```
public static int fib(int n){
    int a,b,t;
    if (n==0 || n==1) {
        return n;
    }
    for (int i=2; i<n; i++){
        t=a+b;
        a=b;
        b=t;
    }
    return t;
}
```