Lecture 13 - Recursion

General practice:

- Return on base case
- Call next iteration of function
- Eliminate duplicate calls with DP/memoization

```
double power(double x, int n) {
  if (n == 0) {
    return 1
  }
  if (n == 1) {
    return x;
  }

  partialRes = power(x, n << 1);

  if (n % 2 == 0) {
    return partialRes * partialRes;
  }
  else {
    return partialRes * partialRes * x; //?? i dont remember
  }
}
```

Summary:

- Recursion relies on mathematical induction
- Simplifies problem solving (fib, linked list traversal)
- Can speed up problem by reducing number of operations
- Can make solving an intracable problem possible

AA BBB solution:

- Base where n=0: none, n=1: none, n=2: AA, n=3: BBB, n=4 = AAAA
- To solve n, call function on n-2, n-3
- Memoization on previously calculated results

```
int[] countWords = new int[n];
countWords[0] = 0;
countWords[1] = 0;
countWords[2] = 1;
countWords[3] = 1;

for (int i = 4; i < n; i++) {
  countWords[i] = countWords[i-2] + countWords[i-3];
}
```