- 알고리즘-algorithm-"생각 사고 훈련"
1. 판단 사고
2. 배열을 이용한 반복 필요

3. 검색방법 – 여러개의 데이터셋에서 특정값 찾기
선형검색 – 연속 데이터 차례대로 검색 – 배열(0 ~ length-1)
n/2 평균 검색 시간
이진검색 – 정렬상태 데이터셋에서
데이터셋 반으로 나눈다
1/2 -> 1/4 -> 1/8
Arrays.sort(배열 자연정렬)
class A implements Comparator<int[]>{
int compare(int[] a , int[] b){
a가 b보다 뒤에 나열되야 한다면1 리턴
a가 b보다 앞에 나열되야 한다면 -1 리턴
a, b 같으면 0 리턴
}
}==> int[][] 정렬 사용
class B implements Comparator<String>{
int compare(String , String){}

}==> String[] 정렬 사용

Arrays.sort(배열 , new B())
Arrays.sort(배열 , new A())

Arrays.binarySearch(배열, 값);

Arrays.toString(1차배열);==> 배열 모든 데이터들 차례대로

4. stack과 queue
List - 순서 유지 컬렉션

stack	queue
Last in First Out 데이터 저장 push / 데이터 삭제 pop – top 1군데 java.util.Stack 클래스 push / pop 메소드	데이터 저장 enqueue- rear 데이터 삭제 dequeue- front First in First Out java.util.Queue 인터페이스 구현 클래스들 Queue<String> q = new LinkedList();

{} {[()]} {[}]--> ?	q.add() q.poll() 은행창구 대기표 업무처리

5. 재귀 recursive

1은 자연수
1+1=2 자연수
2+1=3 자연수

반복문 구현 --> 재귀 변경 구현

```
is자연수(int n){
   if( n > 0) s.o.p("자연수");
   n = n - 1;
   is자연수(n);//100 ~ ... 1
   if(n == 0) return; 꼬리조건
}

main(){
  is자연수(100);
```

5!= 1 ~ 5 누적곱

5! = 5 * 4*3*2*1

5! = 5 * 4!;

4!= 4 * 3!;

3!= 3 * 2!;

2! = 2 * 1;

1! = 1 * 0!;

if(0 같으면) return 1;

0! = 1

1! = 1

2!= 2

int f1(int su){
int result = 1;
for(int i=1 ; i <=su; i++){
result = result * i; i=5 이면 result<-5 * 4 *3 * 2 * 1
}
return result;
}
	int f2(int su){
	if(su == 0 || su == 1) return 1;
	return su * f2(su-1);
	}

12 18 두 수의 최대공약수	
12 약수 – 12를 1~12 숫자 나누어 나머지 0 – 약수 1, 2, 3, 4, 6, 12	18 약수 1 2 3 6 8 18
12 18 공약수들 – 1 2 3 6	
12 18 최대공약수 – 6	
18 – 12 = 6 12 – 6 = 0 차이 계산 0 작은값 최대공약수	
18 % 12 == 0	

fibonacci 수열 1항 ~ 10항

1 1 2 3 5 8 13 21 34 55

int first = 1, second=1;

int i = first+second;

int j = second + i;

int k = i + j;

===6 일 때 fibo 시작=== ===6 일 때 fibo 종료=== ===4 일 때 fibo 시작=== ===4 일 때 fibo 종료=== ===2 일 때 fibo 시작=== ===3 일 때 fibo 시작=== ===3 일 때 fibo 종료=== ===1 일 때 fibo 시작=== 종료값 리턴 ===2 일 때 fibo 시작=== ===5 일 때 fibo 시작=== ===5 일 때 fibo 종료=== ===3 일 때 fibo 시작=== ===3 일 때 fibo 종료=== ===1 일 때 fibo 시작=== ===2 일 때 fibo 시작=== ===4 일 때 fibo 시작=== ===4 일 때 fibo 종료=== ===2 일 때 fibo 시작=== ===3 일 때 fibo 시작=== ===3 일 때 fibo 종료=== ===1 일 때 fibo 시작=== ===2 일 때 fibo 시작=== 6 항의 수열값은 = 8	===6 일 때 fact2 시작=== ===6 일 때 fact2 종료=== ===5 일 때 fact2 시작=== ===5 일 때 fact2 종료=== ===4 일 때 fact2 시작=== ===4 일 때 fact2 종료=== ===3 일 때 fact2 시작=== ===3 일 때 fact2 종료=== ===2 일 때 fact2 시작=== ===2 일 때 fact2 종료=== ===1 일 때 fact2 시작===종료값 리턴 6!(재귀호출)=720

top-down 방식 5! = 5 * 4! 4! = 4 * 3! ==> 이전계산메소드호출 결과 적용 재귀호출-factorial	bottom – up 방식 ==> 같은 항 계산메소드 "여러번" 호출 ==> 반복문 구조 작은값 계산 저장- 재사용

하노이의탑

8퀸

- 잔돈 받기

2000원 내고 물건을 샀다.

1350원짜리 물건이다.

잔돈 받자

650원 – 500원 1개, 100원 1개, 50원 1개

2830원 – 500원 5개 , 300원 1개, 10원 3개

int[] coins = {10, 50, 100, 300, 500}; (잔돈 종류)

int money = 2000; 내가 낸돈

int price = 1350; 상품 가격

잔돈 계산하여 잔돈 갯수를 최소화하여 받자

출력예) 650원 – 500원 1개, 100원 1개, 50원 1개

출력예) 2830원 – 500원 5개 , 300원 1개, 10원 3개