public static boolean searchpair(int[] a,int b){
for (int i = 0; i < a.length; i++) {
for(int j = 0; j < a.length && j != i; j++) {
if (a[i] + a[j] == b)
return true;
}
}
return false;
}
	
	1,n+1,n
	n,n(n+1),n^2
	n^2
	1
	
	1
	
	Either return true or false, not both

Space Complexity: O(n) Time Complexity: O(n^2)	3n^2 + 4n + 3

public static boolean searchtriple(int[] a,int b){
for (int i = 0; i < a.length; i++) {
for(int j = 0; j < a.length && j != i; j++) {
for (int k = 0; k < a.length && k != i && k!= j; k++) {
if (a[i] + a[j] + a[k] == b)
return true;
}
}
}
return false;
}
	
	1,n+1,n
	n,n(n+1),n^2
	n^2,n^2(n+1),n^3
	n^3
	1
	
	1

Space Complexity: O(n) Time Complexity: O(n^3)	3n^3 + 4n^2 + 4n + 3

for (int i = 0; i < n; i++) {
for (int j = 0; j < n; j++) {
double sum = 0;
for (int k = 0; k < n; k++) {
sum += a[i][k] * b[k][j];
}
c[i][j] = sum;
}
}
	1,n+1,n
	n,n(n+1),n^2
	n^2
	n^2, n^2(n+1),n^3
	n^3
	
	n^2

Space Complexity: O(n^2) Time Complexity: O(n^3)	3n^3 + 6n^2 + 4n + 2