Lotus diagram template

								Lotus diagram te
Including the necessary details	and leaving out the unnecessary details when problem solving		Breaking down a complex problem or system into smaller parts that are easier to understand.	T smaller parts can then be solved individually, as they are simpler to work with.				
For example if all sweets cost £0.50 we can ignore that some are red and some are blue	Abstraction	reduce it to a set of essential characteristics.		Decomposition			Algortihmic Thinking	
Maps are a good example	Blueprints (drawings) are a good example	Pattern Recognition can be part of Abstraction						
List needs to be sorted for Binary search to work		Quite a quick method for large list	Abstraction	Decomposition	Algortihmic Thinking	Works with unsorted list (not all lists can be sorted due to how they are saved)		Can be very slow with large lists (think last card in deck)
Split list in half, compare 1 <sup>st</sup> value of "higher" list with target	Binary Search	With small list might be slower than Linear	Binary Search	Algorithms and Computational Thinking	Linear Search	Start at item 0 in list Compare to target	Linear Search	
If target is greater than 1 <sup>st</sup> value then it is in "higher list", if not it is in "lower list"	You get rid of half the list each time you split		Bubble Sort	Merge Sort	Insertion Sort	If it doesn't match move to next value, Repeat until you find target		OK for small lists
	Bubble Sort			Merge Sort			Insertion Sort	