1. Computability determines which problems, or classes of problems can be solved in each model of computation while complexity characterises the behaviour of a [system](https://en.wikipedia.org/wiki/System" \o "System) or [model](https://en.wikipedia.org/wiki/Model_(disambiguation)" \o "Model (disambiguation)) whose components [interact](https://en.wikipedia.org/wiki/Interaction" \o "Interaction) in multiple ways and follow local rules, meaning there is no reasonable higher instruction to define the various possible interactions.
2. Computability theory deals primarily with the question of the extent to which a problem is solvable on a computer. For example, the statement that the [halting problem](https://en.wikipedia.org/wiki/Halting_problem" \o "Halting problem) cannot be solved by a Turing machine is one of the most important results in computability theory, as it is an example of a concrete problem that is both easy to formulate and impossible to solve using a Turing machine. Much of computability theory builds on the halting problem result while [Complexity theory](https://en.wikipedia.org/wiki/Computational_complexity_theory" \o "Computational complexity theory) considers not only whether a problem can be solved at all on a computer, but also how efficiently the problem can be solved. Two major aspects are considered: time complexity and space complexity, which are respectively how many steps does it take to perform a computation, and how much memory is required to perform that computation.
3. Definitions:
4. Set: A set is a well-defined collection of distinct objects, considered as an object in its own right.
5. Power set: The power set (or powerset) of any [set](https://en.wikipedia.org/wiki/Set_(mathematics)" \o "Set (mathematics)) S is the set of all [subsets](https://en.wikipedia.org/wiki/Subset" \o "Subset) of S, including the [empty set](https://en.wikipedia.org/wiki/Empty_set" \o "Empty set) and S itself, variously denoted as P(S).
6. Members of a set: it is any one of the distinct objects that make up that set.
7. Subset: a set A is a subset of a set B, or equivalently B is a superset of A, if A is contained in B.
8. Proper subset: A proper subset of a set A is a subset of A that is not equal to A.
9. Infinite set: Infinite set is a set that is not a finite set. Infinite sets may be countable or uncountable
10. Finite set: finite set is a set that has a finite number of elements. It is a countable set.
11. Unordered pair: an unordered pair or pair set is a set of the form {a, b}, i.e. a set having two elements a and b with no particular relation between them.
12. Union of a set: the union (denoted by ∪) of a collection of sets is the set of all elements in the collection.
13. Intersection of a set: intersection of two sets A and B, denoted by A ∩ B, is the set containing all elements of A that also belong to B (or equivalently, all elements of B that also belong to A).
14. Complement of a set: complement of a set, denoted A', is the set of all elements in the given universal set U that are not in A.
15. Difference of a set: The difference of set B from set A, denoted by A-B, is the set of all the elements of set A that are not in set B.
16. Symmetric difference of a set: symmetric difference, also known as the disjunctive union, of two sets is the set of elements which are in either of the sets and not in their intersection.
17. Answers:
18. B∩C: {0,1,2,3,4,5,8,…,20}

A∩D: {1,2,3,4,5,6,7,8,9}

A∩C: {1,2,3,5,6,9,…}

1. Jk
2. A⊕C: {1,3,5,6,9}

B⊕C: {0,1,2,3,4,5,8,…,20}

C⊕D: {2,4,6,8}

D⊕B: {0,1,2,3,4,5,6,7,8,9,…,20}

1. A∪B: {0,2,3,4,5,6,8,9,…,20}

B∪D: {0,1,2,3,4,5,6,7,8,9,…20}

C∪D: {1,2,3,4,5,6,7,8,9,…}

1. Definitions:
2. Alphabet: An alphabet is a finite set of symbols.
3. Words: a word is a finite sequence of symbols of an alphabet.
4. Length of a word: The number of symbols in a string
5. Substring: A string x is called a substring of another string y if there are strings u and v such that y = uxv. Note that u and v may be an empty string. So a string is a substring of itself.
6. Initial segment:
7. Concatenation of strings: it is the operation of joining character strings end-to-end. For example, the concatenation of "snow" and "ball" is "snowball".
8. Language: it is a set of strings over an alphabet.
9. Answers:
10. L(V): 13
11. VU: concatenationcatetioncona
12. UR: anocnoitetac
13. Cacontatione

∑0: {λ}

∑1: {c,a,c,o,n,t,a,t,I,o,n,e}

∑2: {ca,ac,co,on,nt,ta,at,ti,io,on,ne}

∑3:{cac,aco,con,ont,nta,tat,ati,tio,ion,one}

∑4: {caco,acon,cont,onta,ntat,tati,atio,tion,ione}

∑5: {cacon, acont, conta,ontat,ntati,tatio,ation,tione}

∑6: {cacona,aconta, contat,ontati,ntatio,tation,atione}

∑7: {caconta, acontat, contati, ontatio, ntation, tatione}

∑8: {cacontat, acontati, contatio, ontation, ntatione}

∑9: {cacontati, aacontatio, contation, ontatione}

∑10: {cacontatio, acontation, contatione}

∑11: {cacontation, acontatione}

∑12: {cacontatione}

M={λ,c,a,c,o,n,t,a,t,I,o,n,e,ca,ac,co,on,nt,ta,at,ti,io,on,ne,cac,aco,con,ont,nta,tat,ati,tio,ion,one,caco,acon,cont,onta,ntat,tati,atio,tion,ione,cacon, acont, conta,ontat,ntati,tatio,ation,tione,cacona,aconta, contat,ontati,ntatio,tation,atione,caconta, acontat, contati, ontatio, ntation, tatione,cacontat, acontati, contatio, ontation, ntatione,cacontati, aacontatio, contation, ontatione,cacontatio, acontation, contatione,cacontation, acontatione,cacontatione}

1. S = abracaba
2. ∑0:{λ}

∑1:{a,b,r,a,c,a,b,a}

∑2:{ab,br,ra,ac,ca,ab,ba}

∑3:{abr,bra,rac,aca,cab,aba}

∑4:{abra,brac,raca,acab,caba}

∑5:{abrac,braca,racab,acaba}

∑6:{abraca,bracab,racaba}

∑7:{abracab,bracaba}

∑8:{abracaba}

S={λ,a,b,r,a,c,a,b,a,ab,br,ra,ac,ca,ab,ba,abr,bra,rac,aca,cab,aba,abra,brac,raca,acab,caba,abrac,braca,racab,acaba,abraca,bracab,racaba,abracab,bracaba,abracaba}

1. Jk