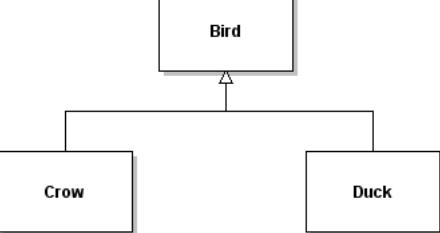


# Canterbury QuestionBank

Field	Value
ID	633573 [created: 2013-06-19 15:29:45, author: crjjrc (xchris), avg difficulty: 2.0000]
Question	Suppose you try to perform a binary search on a 5-element array sorted in the reverse order of what the binary search algorithm expects. How many of the items in this array will be found if they are searched for?
A	5
B	0
*C*	1
D	2
E	3
Explanation	Only the middle element will be found. The remaining elements will not be contained in the subranges that we narrow our search to.
Tags	Contributor_Chris_Johnson, ATT-Transition-ApplyCSspeak, Skill-PureKnowledgeRecall, ATT-Type-How, Difficulty-1-Low, Block-Horizontal-2-Struct_Control, ExternalDomainReferences-1-Low, TopicSimon-Arrays, Block-Vertical-2-Block, Bloom-2-Comprehension, Language-none-none-none, LinguisticComplexity-1-Low, CS2, CodeLength-NotApplicable, TopicWG-Searching-Binary, ConceptualComplexity-2-Medium, Nested-Block-Depth-0-no_ifs_loops

Field	Value
ID	632805 [created: 2013-05-24 02:38:28, author: crjjrc (xchris), avg difficulty: 0.0000]
Question	Which data structure used to implement <code>Set</code> yields the worst performance for <code>Set.contains</code> ?
A	Binary search tree
*B*	Linked list
C	Sorted array
D	Hashtable
Explanation	Implementing <code>Set.contains</code> involves a search of the data structure. A binary search tree and a sorted array are searched in $O(\lg n)$ time, and a hashtable in $O(1)$ , assuming a sane hash function. A linked list is searched in $O(n)$ time.
Tags	Contributor_Chris_Johnson, ATT-Transition-CSspeak_to_Code, ATT-Type-How, Difficulty-1-Low, SkillWG-AnalyzeCode, TopicWG-ADT-Set-Implementations, TopicSimon-AlgorithmComplex-BigO, Block-Horizontal-2-Struct_Control, ExternalDomainReferences-1-Low, Block-Vertical-4-Macro-Structure, Bloom-3-Analysis, Language-none-none-none, LinguisticComplexity-1-Low, CS2, CodeLength-NotApplicable, ConceptualComplexity-2-Medium

Field	Value
ID	635053 [created: 2013-06-17 08:33:10, author: xrobert (xrobert), avg difficulty: 0.5000]
Question	 <pre> classDiagram     class Bird {         &lt;&lt;Bird&gt;&gt;     }     class Crow {         &lt;&lt;Crow&gt;&gt;     }     class Duck {         &lt;&lt;Duck&gt;&gt;     }     Bird &lt; -- Crow     Bird &lt; -- Duck   </pre> <p>The simplified UML diagram above shows the relationships among Java classes Bird, Crow, and Duck.</p> <p>Suppose Bird has a <code>fly(Location place)</code> method, but we want Crows to <code>makeNoise()</code> just before they take off and then behave like other Birds. Assuming Crows have a <code>makeNoise()</code> method, we should</p>