Solutions to second in-class test

Question 1

This is the set of integers whose square is less than 10.

(a) Contained in but not equal to A, as it doesn't contain the negative members of A.

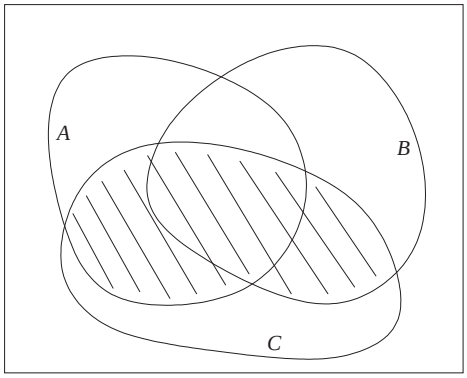
(b) Equal to A.

(c) This is the set of perfect squares below 10, but we are looking for those numbers whose square is below 10. So it is not equal to A

(d) Not equal to A.

(e) Not equal to A, as it contains *all* negative numbers and for most of those the square is much larger than 10.

Question 2



Question 3

(a) We list all Java programs of length 0, then all programs of length 1, and so on. Since there are only finitely many characters allowed in the source code of a Java program, there are only finitely many of a given length. This argument shows we'll get to each program eventually in this manner.

(b) That there are function from $\mathbb{N}$ to $\mathbb{N}$ which can not be implemented as a Java program.

Question 4

(a) This is a function. It is not injective as for a very large `double` the result will be `+infinity` and this will be the result also if the input is `+infinity`. Another reason is that for very large `x` the addition of 1.0 will have no effect at all. Yet another reason is that for very small `x` the result will be 1.0 independent of the input.

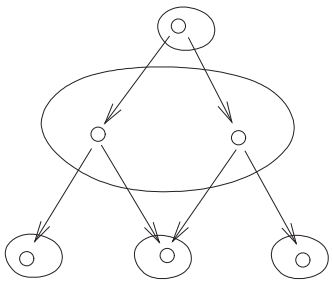
It is not surjective either because for input `-infinity` the answer will be `-infinity` so we can never have the smallest true negative number that is representable as a `double`. (Note that the "infinities" are proper values in a `double`.)

(b) This is a function. It is not injective because of the same (first) reason as in (a). It is not surjective because the smallest positive number will never be twice of any other number.

(c) This is not a function because the method does not return the same value when called with the same argument again.

(a) $\forall w \in W.\ (w, v_1) \in E \iff (w, v_2) \in E$.

(b)



Only the two middle nodes are siblings. Every other node is in a class by itself.

(c) This is all pretty self-evident:

Reflexivity: Of course, every node has the same ancestors as itself.
Symmetry: If v has the same ancestors as w then w has the same ancestors as v.
Transitivity: If v and w have the same ancestors, and w and u have the same ancestors, then v has the same ancestors as u.