CSE 332: Data Structures and Parallelism

Section 10: P/NP

0. Definitions

a) What does P stand for?

Polynomial Time

b) What is NP stand for?

Nondeterministic Polynomial Time

c) What is the definition of P?

The set of problems which can be solved by an algorithm that runs in polynomial time

d) What is the definition of NP?

The set of problems which can be verified by an algorithm that runs in polynomial time.
– Or Equivalently –
The set of problems which can be solved by a nondeterministic algorithm that runs in polynomial time

1. P & NP Membership

a) How would we show that a given problem belongs to the class P?

Find a polynomial time algorithm to solve it

b) How would we show that a given problem belongs to the class NP?

Find a polynomial time algorithm to verify it

2. P & NP Membership

For problems A and B below, show that they belong to both P and NP. Show that problem C belongs to NP.

a) Problem A: Given a list of 2-dimensional points, return true or false to indicate whether some pair of points have a distance of less than 5.

Belongs to P

Algorithm: For each point, find its distance to every other point. If the distance is ever less than 5, return true. After doing this for all points, return false.

Running time: O(n^2)

Belongs to NP

Verification Algorithm: Given a candidate pair of points, verify that those points are both in the list, then verify that their distance is less than 5.

Running time: O(n)

b) Problem B: Given a list of integers, return true or false to indicate whether any items are duplicated (so return true if some value appears at least twice, false if all are distinct).

Belongs to P

Algorithm: Sort the list, then check if any two neighboring items match. Running time: O(n log n)

Belongs to NP

Verification Algorithm: Given a candidate value, check that the value appears at least twice in the list. Running time: O(n)

c) Problem C: Given a weighted graph, a pair of nodes X and Y, and a number k, return true or false to indicate whether there is a path from X to Y with a cost of at least k

Belongs to NP

Verification Algorithm: Given a candidate path, check that it is a valid path (all adjacent nodes in the path form edges in the graph), and check that the sum of those edge weights is at least the value k.

Running time: O(V + E)

3. NP-Hard and NP-Complete Definitions

a) What is the definition of NP-Hard?

The set of problems for which there exists a polynomial time reduction from any NP problem to it

b) What is the definition of NP-Complete?

$NP \cap NPHard$

4. NP-Hard and NP-Complete Membership

a) How do you show that a problem belongs to NP-Hard

Reduce a known NP-Hard problem to it

b) How do you show that a problem belongs to NP-Complete

Show separately that it belongs to both NP-Hard and NP.

Reduce a known NP-Hard to it and find a polynomial time algorithm to verify solutions.

5. Practice

If A polynomial-time reduces to B and B is NP-Hard then A is NP-Hard.

True False

If B is NP-Hard and there exists a polynomial time algorithm for B, then P=NP.

True False

If B is NP-Hard and there does not exist a polynomial time algorithm for B, then P does not equal NP.

True False

If A reduces to B in polynomial time, and B reduces to C in polynomial time, and A is NP-Hard, then C is NP-Hard.

True False

If A reduces to B in polynomial time, and B reduces to C in polynomial time, and A is NP-Complete, then C is NP-Complete.

True False

If A reduces to B in polynomial time, and B reduces to C in polynomial time, and A is in EXP, then C is EXP.

True False

If A reduces to B in polynomial time, and B reduces to C in polynomial time, and A is in P, then C is P.

True False

If A reduces to B in polynomial time, and B reduces to C in polynomial time, and A is in NP, and C is in P, then P=NP.

True False

If A reduces to B in polynomial time, and B reduces to C in polynomial time, and A is in NP-Hard, and C is in P, then P=NP.

True False