Data Structures and Algorithms
INFO 6205
Homework 12
Due: April 21, 2019

Put all your java, compiled class files and documentation files into a zip file named Homework12.zip and submit it via the dropbox on the blackboard before the END of due date. Put your name on all .java files. There will be a short Quiz on this homework.

1. Consider the sequences x = TACGGGTAT and y = GGACGTACG. Assume that the match score is +1, and the mismatch is -1, and gap penalties is -2.

a) Write Java code to use dynamic programming to solve sequence alignment.

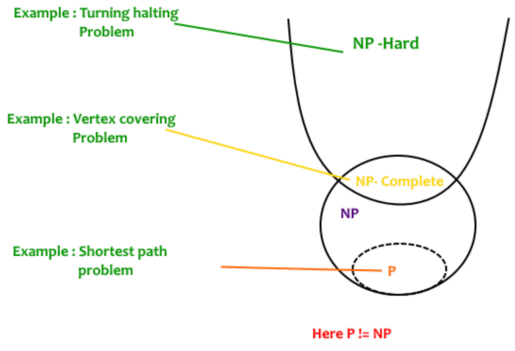
```
Sequence 1: -TACGGGTA-T
Sequence 2: GGAC--GTACG
          T    A    C    G    G    G    T    A    T
     0 - -2 - -4 - -6 - -8 --10 --12 --14 --16 --18
 G | -2 \ -1 \ -3 \ -5 \ -5 \ -7 \ -9 --11 --13 --15
 G | -4 \ -3 \ -2 \ -4 \ -4 \ -4 \ -6 - -8 --10 --12
 A | -6 \ -5 \ -2 \ -3 \ -5 \ -5 \ -5 \ -7 \ -7 - -9
 C | -8 \ -7 | -4 \ -1 - -3 - -5 \ -6 \ -6 \ -8 \ -8
 G |-10 \ -9 | -6 | -3 \  0 \ -2 \ -4 - -6 \ -7 \ -9
 T |-12 \ -9 | -8 | -5 | -2 \ -1 \ -3 \ -3 - -5 \ -6
 A |-14 |-11 \ -8 | -7 | -4 \ -3 \ -2 \ -4 \ -2 - -4
 C |-16 |-13 |-10 \ -7 | -6 \ -5 \ -4 \ -3 | -4 \ -3
 G |-18 |-15 |-12 | -9 \ -6 \ -5 \ -4 \ -5 \ -4 \ -5

Process finished with exit code 0
```

		T	A	C	G	G	G	T	A	T
	0	-2	-4	-6	-8	-10	-12	-14	-16	-18
G	-2	-1	-3	-5	-5	-7	-9	-11	-13	-15
G	-4	-3	-2	-4	-4	-4	-6	-8	-10	-12
A	-6	-5	-2	-3	-5	-5	-5	-7	-7	-9
C	-8	-7	-4	-1	-3	-5	-6	-6	-8	-8
G	-10	-9	-6	-3	0	-2	-4	-6	-7	-9
T	-12	-9	-8	-5	-2	-1	-3	-3	-5	-6
A	-14	-11	-8	-7	-4	-3	-2	-4	-2	-4
C	-16	-13	-10	-7	-6	-5	-4	-3	-4	-3
G	-18	-15	-12	-9	-6	-5	-4	-5	-4	-5

b) What is the optimal alignment path?
TACG
TACG

Follow the graph, the score should be -26(Add each number on the green line).

2. Describe the following:



Relationship of NP-Hard, P, NP and NP-Complete

NP-Hard, provide three examples:

A problem is NP-hard if an algorithm for solving it can be translated into one for solving any NP-problem (non- deterministic polynomial time) problem. NP-hard therefore means "at least as hard as any NP-problem," alt- hough it might, in fact, be harder.

NP-Hard problems are a class of problems which are at least as hard as the hardest problems in NP. Problems in NP-hard do not have to be elements of NP, indeed, they may not even be decision problems.

Examples:

1. An optimization problem - finding the least-cost cyclic route through all nodes of a weighted graph, com-

monly known as the traveling salesman problem.

2. The subset sum problem is an important decision problem in complexity theory and cryptography. There

are several equivalent formulations of the problem. One of them is: given a set (or multiset) of integers, is there a non-empty subset whose sum is zero? For example, given the set {-7,-3,-2,5,8}, the answer is yes because the subset {-3,-2,5} sums to zero.

3. Halting problem

question, so this is a decision problem.

P:

A problem is assigned to the P (polynomial time) class if there exists at least one algorithm to solve that problem, such that the number of steps of the algorithm is bounded by a polynomial in n, where n is the length of the input.

P is the set of yes/no problems that can be solved in polynomial time Examples:

1. Palindrome detection – solvable in linear time.

2. The string matching problem, where we have two strings, if we take p and t. Does p occur as a contiguous

substring of t ? order – O(n2)

3. The greatest common divisor gcd(x, y) of two integers x and y is the largest integer that is a factor of both x and y. For example, gcd(12, 30) = 6.

Two integers x and y are relatively prime if gcd(x, y) = 1.

NP:

In computational complexity theory, NP (nondeterministic polynomial time) is a complexity class used to classify decision problems. NP is the set of decision problems for

which the problem instances, where the answer is "yes", have proofs verifiable in polynomial time.

It is the set of decision problems solvable in polynomial time by a non-deterministic Turing machine.

Examples:

1. The decision version of the travelling salesman problem is in NP.

2. The Boolean satisfiability problem, where we want to know whether a certain formula in propositional logic with Boolean variables is true for some value of the

NP-Complete

1 Decision Problem A is called NP complete if it has following two properties:-

• It belongs to class NP.

• Every other problem in NP can be transformed to P in polynomial time.

Class of decision problems which contains the hardest problems in NP. Each NP-complete problem has to be in NP.

Examples: Vertex cover problem, SAT (Boolean satisfiability problem)

Satisfiability Model, give example

In computer science, the Boolean satisfiability problem (sometimes called propositional satisfiability problem and abbreviated SATISFIABILITY or SAT) is the problem of determining if there exists an interpretation that sat- isfies a given Boolean formula. In other words, it asks whether the variables of a given Boolean formula can be consistently replaced by the values TRUE or FALSE in such a way that the formula evaluates to TRUE. If this is the case, the formula is called satisfiable. On the other hand, if no such assignment exists, the function expressed by the formula is FALSE for all possible variable assignments and the formula is unsatisfiable.

Example:

• The formula "a AND NOT b" is satisfiable because one can find the values a = TRUE and b = FALSE, which make (a AND NOT b) = TRUE. In contrast, "a AND NOT a" is unsatisfiable.

3-SAT, 1-SAT