Home Assignment 4

Jaan Tollander de Balsch

December 11, 2018

Question 1

A kite is a graph on an even number of vertices, say $2n$, in which n of the vertices form a clique and the remaining n vertices are connected in a *tail* that consists of a path joined to one of the vertices of the clique. Given a graph and a goal g, the kite problem asks for a subgraph which is a kite and which contains $2g$ vertices. Prove that kite is NP-complete.

Question 2

Consider two problems:

1) Given a graph $G = (V, E)$ and integer k, find an independent set of size at least k.
2) Given a graph $G = (V, E)$, compute the size of maximum independent set.
3) Given a graph $G = (V, E)$ and integer k, decide whether graph G has an independent set of size k.

Prove that these three problems are equally hard. That is, show that if one of them admits an efficient algorithm, then all of them do.

- TODO: definition of independent set
- TODO: how is it computed
- related problems: clique, vertex cover
- (Cormen et al., 2009, pg. 1102)

Independent-Set(G, k)

- **Input**: Graph $G = (V, E)$ and positive integer k.
- **Output**: Vertices $V' \subset V$ that form an independent set of size k. If none exists returns $V' = \emptyset$.

Has-Independent-Set(G, k)

1) $V' = \text{Independent-Set}(G, k)$
2) **return** is-non-empty(V')

Max-Independent-Set(G)

1) $k = 1$
2) **while** Has-Independent-Set(G, k)
3) ____ $k = k + 1$
4) **return** k

Question 3

In the EXACT 4SAT problem, the input is a set of clauses, each of which is a disjunction of exactly four literals, and such that each variable occurs at most once in each clause. The goal is to find a satisfying assignment if one exists. Prove that EXACT 4SAT is NP-complete.

The *Exact 4-CNF satisfiability* (EXACT 4SAT) problem can be shown to be NP-complete by showing that it reduces to *3-CNF satisfiability* (3SAT) problem which is known to be *NP-complete.* (Cormen et al., 2009, pg. 1082), (Dasgupta, Papadimitriou and Vazirani, 2006, pg. 265)

The reduction works by transforming all the clauses of *four variables* into conjuction of *two clauses* of *three variables.* A clause consisting of four variables can be converted into conjunction of clauses consisting of three variables

$$(a_1 \vee a_2 \vee a_3 \vee a_4) \iff (a_1 \vee a_2 \vee y_1) \wedge (\neg y_1 \vee a_3 \vee a_4).$$

Because the formulas are logically equivalent they have both the same satisfying assignments for variables a_i.

1) The original formula is *unsatisfiable* only if all the variables a_i are false 0. The new formula is also unsatisfiable for any assignment of variables y_j

$$(0 \vee 0 \vee 0 \vee 0) = (0 \vee 0 \vee y_1) \wedge (\neg y_1 \vee 0 \vee 0) = 0.$$

2) The formula is *satisfiable* if one or more of the variables a_i is true 1. The new formula is then also satisfiable for some assignment of variables y_j.

The transformation is *polynomial-time operation.* The transformation algorithm has to simply iterate over all the clauses in the original CNF and convert them into two clauses of three variables. For n clauses this will take $O(n)$ time.

Question 4

Consider the following two problems:

1) **Problem X**: Given an undirected graph $G = (V, E)$ and integer k, find a cut $(C, V - S)$ such that there are at least k edges across the cut, i.e. $|\{(u, v) \in E : u \in S, v \in V - S\}| \geq k$.

2) **Problem Y**: Given a collection of m equations over n variables $x_1, ..., x_n$ under modulo 2:

$$\begin{aligned} x_1 + x_2 &= 1 \mod 2 \\ x_2 + x_3 + x_4 &= 0 \mod 2 \\ &\;\;\ldots \\ x_i + x_j + x_l &= 1 \mod 2 \end{aligned}$$

The goal is to decide whether there is an assignment $\sigma : \{x_1, x_2, ..., x_n\} \to \{0, 1\}$ that satisfies at least k equations.

Part 1: Describe a reduction from Problem X to Problem Y. Since Problem X is NP-hard, this reduction implies that Problem Y is NP-hard.

Part 2: Consider a special case of problem Y where each equation involves at most 2 variables. Present an $O(n + m)$ time algorithm that decides whether all m equations can be simultaneously satisfied.

Does the existence of this algorithm contradict Part 1 where we just proved that Problem Y is NP-hard?

Question 5

This question is taken directly from Ericson's chapter 12. Please use the definitions of universal, uniform and pairwise independent hash functions from his book.

Each of the following questions is worth one point:

a) Describe a set of hash functions that is uniform but no (near-)universal.
b) Describe a set of hash functions that is universal but not (near)uniform.
c) Describe a set of hash functions that is universal but (near-)3-universal.
d) Describe a set of hash functions that is uniform but not pairwise independent.
e) Describe a set of hash functions that is pairwise independent but not (near-)uniform.
f) Describe a set of hash functions that is universal but not pairwise independent.
g) Describe a set of hash functions that is pairwise independent but not (near-)uniform.
h) Describe a set of hash functions that is universal and pairwise independent but not uniform, or prove no such set exists.

Note: For this particular question, an answer without proof will not receive any point.

Question 6

In a student chess competition, there are n students (we call these students by numbers $1, 2, 3, ..., n$) who played one-on-one decisive matches against each other (i.e. each game only ends in win/loss but not a draw). There were m matches in total.

The competition is over. To encourage a more fierceful competition among students in the next year, an evil coach wants to group these students into two groups: He calls the groups, the elite, and the sucker groups respectively. The coach wants to group the students in such a way that the grouping is as *consistent* with the competition results as possible. In particular, each match record of the form *i lost to j* is said to be consistent with the grouping if student i is placed in the sucker group and student j is placed in the elite group. The coach aims at maximizing the number of consistent match results.

Part 1: A randomized algorithm picks a partition into two groups at random. Prove that the expected number of consistent matches is at least 1/4 of the total number of matches.

Part 2: Describe an efficient deterministic algorithm that partitions students into two groups such that the number of consistent matches is at least 1/4 of the total number of matches.

Part 3: Now, instead of partitioning into only two groups, the coach wants to rank the students (no ties are allowed). A ranking is said to be consistent with the match *i lost to j* if the rank of student i is worse than the rank of student j. The objective of the coach is to produce a ranking that is as consistent as possible.

A randomized algorithm picks a random ranking, i.e. pick a random permutation σ and uses this as a ranking. Prove that the expected number of consistent match results is at least 1/2 of the total number of matches.

Part 4: Describe a deterministic algorithm that, on an input, produces a ranking that is always consistent with at least 1/2 of the total number of matches.

References

Cormen, T.H., Leiserson, C.E., Rivest, R.L. and Stein, C., 2009. *Introduction to algorithms.* MIT press.

Dasgupta, S., Papadimitriou, C.H. and Vazirani, U., 2006. *Algorithms.* 1st ed. New York, NY, USA: McGraw-Hill, Inc.