Algorithmics and complexity

TD 5/7 – Théorie de la complexité

Training exercises

Note: The solution elements given here are not complete. Their purpose is only to guide you. We encourage you to write a proper answer, as you would do for the exam. If any question remains, feel free to ask your tutorial supervisor for help.

Exercise 1 : SUBSETSUM (2018-2019 exam)

The following two problems are considered :

BI-PARTITION
Entrée : A set SE of numbers.
Question : Is there a subset $F \subseteq E$ such that $\displaystyle\sum_{x \in F} x = \frac{1}{2} \sum_{x \in E} x$?

SUBSETSUM
Entrée : A set S of numbers and a number t
Question : Is there a subset $T \subseteq S$ such that $\displaystyle\sum_{x \in T} x = t$?

Question 1

Show that SUBSETSUM is in NP.

Solution elements :

You have to write in Python a **polynomial** algorithm to check a solution T (you can write one of complexity $\mathcal{O}(|S| * |T|)$ or $\mathcal{O}(|T|)$ depending on the data structure you use for T).

Question 2

Knowing that BI-PARTITION is NP-complete, show that SUBSETSUM is also NP-complete.

Solution elements :

Be careful to the reduction direction : you have to take an instance E of BI-PARTITION and build an instance of SUBSETSUM using $S = E$, $t = \frac{1}{2} \sum_{x \in E} x$. Remember to check that the transformation of (E) into (S, t) is polynomial (linear in $|E|$). Then show that if F is a solution to the BI-PARTITION instance E, then we can build a solution T to the SUBSETSUM instance (S, t) and, conversely, if T is a solution to the SUBSETSUM instance (S, t), then we can build F solution to BI-PARTITION instance E (in this simple case, $F = T$).

You have shown that BI-PARTITION $<_p$ SUBSETSUM. As BI-PARTITION is NP-complete, then SUBSETSUM is NP-hard.

But since it is also NP, it is NP-complete.

Exercise 2 : Dominating Set problem

A dominating set in an undirected graph $G = (V, E)$ is a subset $D \subset V$ such that every vertex outside of D has a neighbor in D.

In this exercise we consider the following decision problem : For a given number k, is there a dominating set of size smaller than k ?

Question 1

Formalize the problem of the dominating set.

Solution elements :

Entries :

- A graph $G = (V, E)$ where V is the set of vertices and $E \subseteq V \times V$ is the set of G edges (E is symmetrical).
- $k \in \mathbb{N}$ a natural number

Question : Is there a subset $D \subset V$ such that :

- D is a dominant set of G, i.e. $\forall v \in V \setminus D, \exists u \in D, (u, v) \in E$
- the subset D is of size less or equal to k.

Question 2

Using the Set Cover problem, show that the decision problem of finding a dominating set is NP-complete.

Hint : build a graph $G' = (V, E)$ where :

- $V = I \cup U$,
- $E = \underbrace{\{(i, j), i, j \in I \wedge i \neq j\}}_{E_1} \cup \underbrace{\{(i, u), i \in I, u \in S_i\}}_{E_2}$; This way (V, E_2) is a bipartite graph, (I, E_1) is a clique and (U, E) is an independent set.

Solution elements :

The verification of a solution to Dominant is polynomial. It checks that each node of the graph v in V is close to a node u in the dominant D, i.e. $(u, v) \in E$ (we can use an adjacency matrix, so we run $|V|$ times a search among $|D|$ nodes : $O(|V||D|)$), so Dominant is in NP. (see the corresponding practical exo in Notebooks)

The reduction of Set cover to Dominant is clearly polynomial (quadratic). Then we must check the equivalence between positive instances. Source (Wikipedia) : `https://en.wikipedia.org/wiki/Dominating_set#L-reductions`

- $\Rightarrow$ if $C = \{S_i, i \in D\}$ is a solution to Set Cover with $D \subseteq I$, then D is a dominant in G : first for each $u \in U$ there is an $i \in D$ such that $u \in S_i$ and by construction, u and i are adjacent; so u is dominated by i. And since I is a clique, every $i \in I$ is adjacent to D as $D \subseteq I$.
- $\Leftarrow$ the opposite direction is important and more complicated. If D is a G dominant. So we have $D \subseteq U \cup I$. It is possible to build a new dominant X such that $|X| \leq |D|$ and $X \subseteq I$: simply by replacing each $u \in D \cap U$ by a neighbor $i \in I$ of u. So, $C = \{S_i, i \in D\}$ is a solution to the Set Cover with $|C| = |X| \leq |D|$.

Set Cover is NP-Complete, so Dominant is NP-Hard. In addition, checking a Dominant solution is polynomial (is in NP), so Dominant is NP-complete.

Exercise 3 : Hamiltonian Chain

We suppose that the following problem, called Hamiltonian Cycle, is NP-complete.

Input : **Undirected** graph $G = (V, E)$

Question : Does G contain a Hamiltonian cycle?

Reminder : a Hamiltonian cycle is a path where each vertex is visited exactly once, except for the starting vertex which is also visited at the end.

Let Hamiltonian Chain be the following problem.

Input : An undirected graph $G = (V, E)$ with two distinct vertices u and v.

Question : Does G contain a Hamiltonian chain between u and v?

Reminder : A Hamiltonian chain is a path passing once and only once through each vertex of the graph. If a Hamiltonian cycle can be transformed into a Hamiltonian chain by removing an edge of the cycle, a Hamiltonian chain can only be transformed into a Hamiltonian cycle if its two endpoints are adjacent.

Question 1

Show that the HAMILTONIAN CHAIN problem is NP.

Solution elements :

The problem HAMILTONIAN CHAIN is in NP, because given a string, we can check in polynomial time $\mathcal{O}(|V|^2)$ if it passes once and only once by each vertex of the graph. You can simply write it in Python using an adjacency matrix to represent the graph.

Question 2

Show that the HAMILTONIAN CHAIN problem is NP-complete. To do this, we reduce the HAMILTONIAN CYCLE problem into the HAMILTONIAN CHAIN problem. Let $\mathcal{I} = \langle G = (V, E)\rangle$ be an instance of the HAMILTONIAN CYCLE problem. Let us transform this instance into an instance $\mathcal{I}' = \langle G' = (V', E'), u, v\rangle$ of the HAMILTONIAN CHAIN problem in the following manner :

— u is a vertex from V

— $V' := V \cup \{v\}$ such that v is a new vertex which does not belong to V

— $E' := E \cup \{(v, l) : l$ is a neighbour of u in $G\}$

Continue the proof . . .

Solution elements :

Such transformation can be done in polynomial time as we only copy G while adding one vertex and less then $|E|$ edges.

The proof requires two directions :

— If there is a Hamiltonian chain P in G' between u and v, so there is Hamiltonian cycle C in G. As P has the form $P = (u, l_1, \ldots, l_{n-1}, v)$. the cycle $C = (u, l_1, \ldots, l_{n-1}, u)$ is Hamiltonian in G.

— If there is a Hamiltonian cycle C in G, so there is a Hamiltonian chain P in G'. Without loss of generality, one can begin the cycle in u thus $C = (u, l_1, \ldots, l_{n-1}, u)$. We construct the chain $P = (u, l_1, \ldots, l_{n-1}, v)$ in the graph G'. This chain is Hamiltonian : it passes once and only once through each vertex of G and then visits v last.

So HAMILTONIAN CYCLE $\leq_p$ HAMILTONIAN CHAIN and HAMILTONIAN CHAIN is NP-complete.

Question 3

Chevaliers de la table ronde : Given n knights, and knowing every pair of ferocious enemies among them, is it possible to place them around a circular table so that no pair of enemies are side by side ? What is this problem and what is its complexity class ?

Solution elements :

It is a practical case of the HAMILTONIAN CYCLE problem (as the table is circular).