Exercises NP-completeness

Exercise 1 Knapsack problem

Consider the KNAPSACK PROBLEM. We have n items, each with weight a_j ($j = 1, ..., n$) and value c_j ($j = 1, ..., n$) and an integer B. All a_j and c_j are positive integers. The question is to find a subset of the items with total weight at most B and maximal value.

a) Formulate the decision problem corresponding to KNAPSACK.

b) Show that KNAPSACK belongs to NP.

c) Show that KNAPSACK is NP-complete.

(a) The decision problem corresponding to KNAPSACK is defined as follows. Given an instance of KNAPSACK and a threshold y, does there exist a valid solution of KNAPSACK with objective value $\geq y$?

(b) A 'yes' solution can be encoded in an array in $O(n)$ space: just put a '1' if the item is present in the solution, and a '0' otherwise. Checking the feasibility requires adding the sizes of the items in the solution and comparing this to B; this can be done in $O(n)$ time. Checking whether the answer is 'yes' requires adding the values of the items in the solution and comparing this to y; this can be done in $O(n)$ time.

(c) Consider any instance of SUBSET SUM; it consists of t nonnegative integers $b_1, \ldots, b_t$ and a target value Q. We construct the following special instance of KNAPSACK: each integer b_j ($j = 1, \ldots, t$) corresponds to an item j with size $a_j = b_j$ and value $c_j = b_j$; we choose the size B of the knapsack equal to Q. Remark here that, since the sizes and values of the items are equal, it is optimal to fill the knapsack as much as possible. To obtain the decision variant, we add the threshold y and make it equal to Q. We will show that there exists a feasible solution to the instance of SUBSET SUM if and only if there exists a feasible solution to the instance of KNAPSACK with value $\geq y$.

First suppose that there exists a subset S of $\{1, \ldots, t\}$ such that

$$\sum_{j \in S} b_j = Q.$$

Consider the solution of KNAPSACK in which we put the items in S in the knapsack. Since

$$\sum_{j \in S} a_j = \sum_{j \in S} b_j = Q = B \quad \text{and} \sum_{j \in S} c_j = \sum_{j \in S} b_j = Q = y,$$

we find that this solution is a 'yes' solution of the decision variant of KNAPSACK.

Suppose now that we have a 'yes' solution of the decision variant of KNAPSACK; let S be the corresponding subset of $\{1, \ldots, t\}$. Then we know that

$$\sum_{j \in S} a_j = \sum_{j \in S} b_j \leq B = Q \quad \text{and} \sum_{j \in S} c_j = \sum_{j \in S} b_j \geq y = Q,$$

which implies that $\sum_{j \in S} b_j = Q$, and hence we can find a 'yes' solution for SUBSET SUM by taking the integers b_j for which $j \in S$.

Exercise 2. NP-completeness of Vertex Cover
We are given an undirected graph (V, E). A vertex cover is a subset $W \subseteq V$ such that for each $(v, w) \in E$ we have $v \in W$ or $w \in W$. We consider the following problem.

> VERTEX COVER
> **Instance:** Undirected graph $G = (V, E)$, integer K.
> **Question:** Does G have a vertex cover of at most K vertices?

a) Show that VERTEX COVER belongs to the class NP.

b) Proof that the VERTEX COVER problem is NP-complete by a reduction from INDEPENDENT SET.

a) A solution is a subset V' of the nodes and can be encoded in $O(n)$. Checking if a solution is a 'yes'-solution, you have to check for each edge if it has an endpoint in V', this requires $O(n^2)$ time.

b) Let graph $G = (V, E)$ and integer K be an instance of INDEPENDENT SET. The question is if there is a independent set of size a least K. We define the following instance G', K' of VERTEX COVER. $G' = G$ and $K' = n - K$. You can prove that G has an independent set of size at least K if and only of G' has a vertex cover of size at most K' by showing that in the graph G, W is an independent set G if and only $V \setminus W$ is a vertex cover. (detail to be filled in).

Exercise 3 Hamilton

a) You first have to show that HAMILTONCYCLE is in $\mathcal{NP}$ (you have to write down the details yourself). Then we can prove that HAMILTONCYCLE is $\mathcal{NP}$-complete by a reduction from HAMILTONPATH. Let graph $G = (V, E)$ be an instance of HAMILTONPATH Construct the following, instance of HAMILTONCYCLE. The graph G' consists of a copy of G, where we add node v_0 and an edge from v_0 to every other node $v \in V$. The idea is that if there is an Hamiltonian path in G then this corresponds to a Hamilton cycle by connecting the end nodes through v_0.

   Suppose G is a yes instance in HAMILTONPATH. Then we can construct a tour in G' by connecting the end points of the path through v_0.
   Now suppose G' is a yes instance in HAMILTONCYCLE. Construct a Hamilton path by removing v_0 and the edges connected to v_0. Hence the answer to Hamilton path is also yes.

b) You first have to show that HAMILTONPATH is in $\mathcal{NP}$ (you have to write down the details yourself). Then we can prove that HAMILTONPATH is $\mathcal{NP}$-complete by a reduction from HAMILTONCYCLE. Let graph $G = (V, E)$ be an instance of HAMILTONCYCLE. Construct the following, special instance of HAMILTONPATH. We start with $G' = G$, so $V' = V$ and $E' = E$. The idea is that we cut the tour into pieces such that we get a Hamiltonpath. Select a node $v \in V$. We replace this node by four nodes: v_1', v_2', v_1'', v_2''; we replace each edge $\{v, w\} \in E$ by the edges $\{v_1', w\}$ and $\{v_1'', w\}$, and finally we add the edges $\{v_1', v_2'\}$ and $\{v_1'', v_2''\}$.

Suppose G is a yes-instance in HAMILTONCYCLE. Then you can find a Hamiltonpath by splitting the cycle in node v; you start with v_2', v_1', follow the cycle, and end with v_1'', v_2''.
Now, suppose G' is a yes-instance in HAMILTONPATH, hence there exists a Hamiltonpath in G'. Since the nodes v_2' and v_2'' only have one neighbour these must be end points. the Hamiltonpath starts with v_2', v_1' and ends with v_1'', v_2''. You finds a Hamiltoncycle in G by following the Hamiltonpath from v_1' to v_1'', and then replacing v_1' and v_1'' by v.

Exercise 4 Task scheduling
We consider the following scheduling problem. There is one machine and a set of n tasks, $J_1, \ldots, J_n$. Each task J_i $(1 \leq i \leq n)$ has a processing time p_j, a profit w_j, and a deadline d_j. We must schedule the tasks on the machine, such that the machine carries out at each moment at most one task; tasks run without interruption for p_j time. Tasks that are complete before their deadline give a profit of w_j; other tasks give a profit 0. Suppose a target profit W is given. Show that the problem to decide if a schedule with profit at least W is NP-complete.

Consider the case where $d_j = d$ for all $j = 1, \ldots, n$. If you look at this problem carefully, then you see that it is just the decision variant of the KNAPSACK problem: the processing time corresponds to the size, and the size of the knapsack is equal to d. Now it is straightforward to give a reduction from KNAPSACK.

Exercise 5 Parcels and two trucks
A company has two trucks, and must deliver a number of parcels to a number of addresses. They want both drivers to be home at the end of the day. This gives the following decision problem.

> **Instance:** Set V of locations, with for each pair of locations $v, w \in V$, a distance $d(v, w) \in \mathbb{N}$, a starting location $s \in V$, and an integer K.
> **Question:** Are there two cycles, that both start in s, such that every location in V is on at least one of the two cycles, and both cycles have length at most K?

Show that this problem is NP-complete.

If there is only one driver, then the problem above boils down to the decision variant of the Traveling Salesman Problem (TSP). This gives us the idea to use a reduction from TSP or Hamiltonian Cycle. To be sure that the distances will become non-negative integers, we use a reduction from Hamiltonian Cycle so that in the above problem we can define the distances ourselves, depending on the presence of an adge. The only thing we have to do is to keep the other driver busy, which is easily taken care of by including an additional vertex with distance $K/2$ from s and distance $K + 1$ to all other addresses.

Hence, we find the following reduction. Given any instance of Hamiltonian Cycle with n vertices, construct the following special instance of the above problem. Copy the graph and make the distance $d(v, w)$ equal to 2 if there is an edge in the graph, and 4, otherwise; label one of these vertices as s. Add one vertex 0 such that $d(s, 0) = d(0, s) = n$ and $d(v, 0) = d(0, v) = 2n + 1$ for all $v \neq s$. The threshold K is equal to $2n$.

You have to fill in the remaining details yourself.

Exercise 6 Partition
Subset Sum is defined as follows: given t nonnegative integers $b_1, \ldots, b_t$ and a nonnegative integer Q, does there exist a subset S of $\{1, 2, \ldots, n\}$ such that

$$\sum_{j \in S} b_j = Q?$$

We assume that it is given that Subset Sum is $\mathcal{NP}$-complete.

Partition is defined as follows: given n nonnegative integers $a_1, \ldots, a_n$, does there exist a subset S of $\{1, 2, \ldots, n\}$ such that

$$\sum_{j \in S} a_j = (\sum_{j \in \bar{S}} a_j)?$$

Here $\bar{S} = \{1, 2, \ldots, n\} \setminus S$.

a) Show that Partition is in $\mathcal{NP}$.

b) Prove that Partition is $\mathcal{NP}$-complete by a reduction from Subset Sum.

c) Suppose that we have a instance of PARTITION where the cardinality n of the set of numbers is even. Prove that PARTITION remains $\mathcal{NP}$-complete if we require that the subset S contains exactly $n/2$ elements.

d) * The problem EVEN-ODD PARTITION is defined as follows: given a set A of $2r$ nonnegative integers‘ $\{a_1, \ldots, a_{2r}\}$ with $a_i \geq a_{i+1}$ $(i = 1, \ldots, 2r-1)$, does there exist a subset S of $\{1, 2, \ldots, 2r\}$ such that $\sum_{j \in S} a_j = \sum_{j \in \bar{S}} a_j$, where S contains exactly one element from $\{2i-1, 2i\}$ for every $i = 1, \ldots, r$? Prove that EVEN-ODD PARTITION is $\mathcal{NP}$-complete.

a) This is almost the same as Excercise (1b). A ‘yes’ solution consists of a subset S of $\{1, \ldots, n\}$; this subset can be encoded in an array in $O(n)$ space by putting $A[j] = 1$ if $j \in S$ and $A[j] = 0$, otherwise. Checking whether the answer is ‘yes’ requires adding up the integers, the index of which is in S and $\sum_{j=1} a_j/2$; this can be done in $O(n)$ time.

b) Consider any instance of SUBSET SUM; it consists of t nonnegative integers $b_1, \ldots, b_t$ and a target value Q. Define

$$\bar{Q} = \sum_{j=1}^{t} b_j - Q.$$

The idea is that we construct a special instance of PARTITION that consists of the values present in the instance SUBSET SUM with two additional values $M - Q$ and $M - \bar{Q}$, where M is a big value such that $(M - Q) + (M - \bar{Q}) > M$. If we have a ‘yes’ solution to PARTITION, then we cannot have both additional values in the same subset. Hence, the remaining integers (which are the integers from the instance of SUBSET SUM) must be partitioned into two subsets, such that one has sum Q and the other one has sum $\bar{Q}$.

Working out the details: Construct the special instance of PARTITION that consists of $n = t + 2$ integers $a_1, \ldots, a_{t+2}$. Define $a_j = b_j$ $(j = 1, \ldots, t)$; define $a_{t+1} = M - Q$ en $a_{t+2} = M - \bar{Q}$. Hence, $\sum_{j=1}^{t+2} a_j = 2M$, which implies that you have to find a subset S of $\{1, \ldots, t+2\}$ such that $\sum_{j \in S} a_j = M$.

Suppose that there exists a subest T of $\{1, \ldots, t\}$ such that

$$\sum_{j \in T} b_j = Q.$$

Then choose S equal to $T \cup \{t+1\}$; a quick computation shows that S leads to ‘yes’ to PARTITION.

Conversely, suppose that there exists a subest S of $\{1, \ldots, t+2\}$ such that

$$\sum_{j \in S} a_j = M.$$

Since $a_{t+1} + a_{t+2} > M$, S contains one of the indices in $\{t+1, t+2\}$. Suppose that S contains index $t+1$ (consider $\bar{S}$, otherwise. Then take $T \leftarrow S \setminus \{t+1\}$, and we find that

$$\sum_{j \in T} b_j = \sum_{j \in S} a_j - a_{t+1} = M - (M - Q) = Q,$$

which implies that the answer to Subset Sum is 'yes' as well.

c) We call Partition with the extra requirement that S contain exactly $\frac{n}{2}$ elements Partition with equal cardinality. This problem is in $\mathcal{NP}$; similar to partition. We now give a reduction from Partition to Partition with equal cardinality.

Consider instance of Partition is given by n numbers $b_1, \ldots, b_n$. Construct an instance of Partitie with equal cardinality by adding n number $b_{n+1} = b_{n+2} = \ldots b_{2n} = 0$. It is easy to show that we have a 'yes'-instance in Partition if and only if we have a 'yes'-instance in Partition with equal cardinality.

d) It is readily verified that Even-Odd Partition belongs to the class$\mathcal{NP}$. We will show thatPartition is a special case of Even-Odd Partition. Take any instance of Partition; suppose it consists of n nonnegative integers $a_1, \ldots, a_n$.

The idea behind the reduction is that we choose the difference between the two integers in pair i equal to a_i ($i = 1, \ldots, n$); choosing the larger of the two implies then that i will be included in the subset S.

We construct the special instance of Even-Odd Partition consisting of $2n$ integers $b_1, \ldots, b_{2n}$, such that $b_{2i-1} - b_{2i} = a_i$. Since we need that $b_1 \geq b_2 \geq \ldots \geq b_{2n}$, we choose $b_{2n} = 1$ and compute the remaining b_j ($j = 1, \ldots, 2n-1$) recursively according to the following formulas

$$b_{2i-1} = b_{2i} + a_i \text{ for all } i = 1, \ldots, n,$$

$$b_{2i} = b_{2i+1} \text{ for all } i = 1, \ldots, n-1.$$

We will show for this instance that 'yes' to Partition leads to a 'yes' to Even-Odd Partition and vice-versa. Suppose that the subset S of $\{1, \ldots, n\}$ leads to a 'yes' to Partition. Define A_1 as the set containing index $2j-1$ if $j \in S$ and $2j$ if $j \neq S$, for $j = 1, \ldots, n$. This set leads to a 'yes' to Even-Odd Partition.

Conversely, given a set A_1 leading to a 'yes' to Even-Odd Partition you can construct a subset S of $\{1, \ldots, n\}$ that leads to 'yes' to Partition by putting all indices j in S for which A_1 contains index $(2j-1)$.