Algorithmic Lower Bounds - Assignment 1

Professor: Mohammad T. Hajiaghayi

Problem 1. For each of the following problems, either show that the problem is in P by giving a polynomial-time algorithm or show that the problem is NP-hard by reducing from 3-Partition, its variants 3-Dimensional Matching[1], or Numerical 3-Dimensional Matching[2].

(a) Given a set of numbers $A = \{a_1, \cdots, a_{2n}\}$ that sum to $t \cdot n$, find a partition of A into n sets $S_1, \cdots, S_n$ of size 2 such that each set sums to t.

(b) Given a set of numbers $A = \{a_1, \cdots, a_{2n}\}$ that sum to $t \cdot n$, find a partition of A into n sets $S_1, \cdots, S_n$ of any size such that each set sums to t.

(c) Given a set of numbers $A = \{a_1, \cdots, a_{2n}\}$ and a sequence of target numbers $\langle t_1, \cdots, t_n \rangle$, find a partition of A into n sets $S_1, \cdots, S_n$ of size 2 such that for each $i \in \{1, \cdots, n\}$, the sum of the elements in S_i is t_i.

Solution. (a) Create a graph with one vertex for each input number. For each pair of numbers a_i, a_j check whether $a_i + a_j = t$. If so, add an edge to the graph. Otherwise, there should be no edge between a_i and a_j. Hence, each edge represents a possible group in the partition. Next, run a matching algorithm. If the result is a perfect matching, we can construct the corresponding partition by creating one group for each edge in the matching. Because we start with a matching, each a_i can belong to at most one group in the corresponding partition. And because the matching that we start with is perfect, we are guaranteed to haven groups of size 2. The converse is also true. Suppose that we have a partition satisfying the problem constraints. For each group $\{a_i, a_j\}$ in the partition, we are guaranteed that $a_i + a_j = t$, so the corresponding edge (a_i, a_j) must exist. Hence, we can add it to the matching. Because we started with a partition, no two edges in the matching share an endpoint. And because the number of groups is n while the number of vertices is $2n$, we know that the matching constructed in this fashion must be perfect.

(b) Reduce from standard 3-Partition (the variant where any number of numbers is allowed to belong to a single group). Let $a_1, \ldots, a_{3n}$ be the groups of input numbers. Define a new sequence of numbers $b_1, \ldots, b_{4n}$ as follows:

$$b_i = \begin{cases} a_i & \text{if } i \leq 3n \\ t & \text{otherwise} \end{cases}$$

[1]see https://en.wikipedia.org/wiki/3-dimensional_matching

[2]see https://en.wikipedia.org/wiki/Numerical_3-dimensional_matching

Suppose that we are given a partition of these numbers into $2n$ groups that sum to t. Clearly, $b_{3n+1}, \ldots, b_{4n} = t$, so any group containing one of those nnumbers cannot contain any other numbers. Hence, the remaining n groups must contain all $3n$ numbers from the original 3-Partition instance. Furthermore, each group must sum to t. Hence, the assignment of those numbers to the remaining n groups tells us how to solve the original 3-Partition problem.

(c) Reduction from Numerical 3-Dimensional Matching. Given $A = \{a_1, \ldots, a_n\}$, $B = \{b_1, \ldots, b_n\}$, and $C = \{c_1, \ldots, c_n\}$, with target sum t, we define the new numbers as follows:

$$d_{2(i-1)+1} = 2a_i + 0$$

$$d_{2(i-1)+2} = 2b_i + 1$$

And the target values are:

$$q_i = 2(t - c_i) + 1$$

Suppose that we have a sequence of groups $S_1, \ldots, S_n$ satisfying the desired constraints. By examining the targets modulo 2, we can see that each group must contain exactly one number that is equivalent to 1 mod 2. By construction, only the numbers $d_{2(i-1)+2} \equiv 1 \mod 2$. Therefore, each group must contain exactly one number $d_{2(i-1)+2} = 2b_i+1$, and the other number in each group must be some $d_{2(j-1)+1} = 2a_j+0$. So for the k^{th} group, we must have:

$$d_{2(i-1)+2} + d_{2(j-1)+1} = q_k \Rightarrow 2b_i + 1 + 2a_j + 0 = 2(t - c_k) + 1 \Rightarrow a_j + b_i + c_k = t$$

Which is precisely what we wanted.

Conversely, suppose that we have a solution to the original Numerical 3-Dimensional Matching instance. Then for each group $\{a_i, b_j, c_k\}$, we set $S_k = \{d_{2(i-1)+1}, d_{2(j-1)+2}\}$. We are guaranteed that $a_i + b_j + c_k = t$, so we have:

$$\sum_{x \in S_k} x = 2a_i + 0 + 2b_j + 1 = 2(a_i + b_j) + 1 = 2(t - c_k) + 1 = q_k$$

Problem 2. Give a direct reduction from 3-Partition to Partition.[3]

Solution. Let $a_1, \ldots, a_{3n}$ be the multiset of numbers to partition, and let T be the target sum for each group. For each number a_i and each possible group $k \in \{0, \ldots, n-1\}$, we add the following number to our Subset-sum instance:

$$x_{i,k} = 1 \cdot (Tn)^{n+i} + a_i \cdot (Tn)^k$$

The target number we aim for in a Subset-Sum problem would be:

$$T' = \sum_{i=1}^{3n} 1 \cdot (Tn)^{n+i} + \sum_{k=0}^{n-1} T \cdot (Tn)^k$$

Consider the values mod (Tn). Clearly, T' mod $(Tn) = T$, and for $k \neq 0$, $x_{i,k}$ mod $(Tn) = 0$. So in order to get our target sum, we need to use a subset of the numbers $x_{1,0}, \ldots, x_{n,0}$ that sums to T mod (Tn). By construction, this is equivalent to finding a subset of the numbers $a_1, \ldots, a_{3n}$ that sums to T, and then using the corresponding numbers $x_{i,0}$ in our Subset-Sum problem. A similar argument shows that, for any $k \in \{0, \ldots, n-1\}$, we must pick numbers $x_{i_1,k}, \ldots, x_{i_q,k}$ such that $a_{i_1} + \ldots + a_{i_q} = T$. Furthermore, if we examine the sum mod $(Tn)^{n+i+1}$ for each $i \in \{1, \ldots, 3n\}$, it is clear to see that for each number $i \in \{1, \ldots, 3n\}$, we can pick only one $x_{i,k}$ to belong to our subset sum. Hence, if we can find a subset of numbers that sums to the target, we know that there must exist a partition of $a_1, \ldots, a_{3n}$ into n groups, each of which sums to T.

Next, we wish to convert our reduction to Subset-Sum into a reduction to Partition. The sum of all numbers in our problem is

$$\begin{aligned} Q &= \sum_{i=1}^{3n} \sum_{k=0}^{n-1} x_{i,k} \\ &= \sum_{i=1}^{3n} \sum_{k=0}^{n-1} (1 \cdot (Tn)^{n+i} + a_i \cdot (Tn)^k) \\ &= \sum_{i=1}^{3n} n \cdot (Tn)^{n+i} + \sum_{k=0}^{n-1} (Tn) \cdot (Tn)^k \end{aligned}$$

To ensure that we find a subset that sums to T', we add one extra number $Q - 2T'$. (Note that because Q is very large in comparison to T', this new number will not be negative.) With this extra number, the new total becomes $2Q - 2T'$, so a solution to the Partition problem must make both halves sum to $Q - T'$. One of those halves must contain the extra number $Q - 2T'$, so the set of all other numbers in that half must sum to $(Q - T') - (Q - 2T') = T'$, which is precisely what we wanted.

[3] Hint: First reduce directly from 3-Partition to Subset-Sum, then modify the proof to work with Partition.

Problem 3. In the connected bisection problem, given a graph $G = (V, E)$ with n vertices, one needs to decide if V can be partitioned into two sets, each of size $n/2$ such that each part induces a connected subgraph. Show that this problem is NP-hard.

Solution. We give a reduction from 3-dimensional matching to the connected bisection problem. Consider an instance of the 3-dimensional matching problem that we are given sets X, Y, Z each of size n, and a set $T \subseteq X \times Y \times Z$ of triplets, and we want to decide whether or not, there is a matching $M \subseteq T$, i.e., $|M| = n$ and each element of X, Y, Z occurs in exactly one triple of M. A bipartite view of the 3-dimensional matching problem is as follows: We construct a graph $G = (A, B, E)$ where we have a vertex in A for each of the elements in $X \cup Y \cup Z$ and we also have a vertex in B for each of the triplets in T, and for each triplet $t = (x, y, z)$, we connect its vertex in B to the vertices of x, y and z in A. The goal is to pick a matching which is a subset M of vertices in B such that each vertex in A is neighbor to exactly one vertex in M. Given this instance, we construct graph G' as follows:

We add two vertices a and b and connect them to each vertex in B. Let

$$n_a = (3n + 1)n^3 + 5n - |T| \, ,$$

and

$$n_b = n^3 \, .$$

We add a path of length n_a and connect it to the vertex a. Also, for each vertex $v \in \{b\} \cup A$, we add a path of length n_b and connect it to v. The total number of vertices in graph G' is

$$n' = 2 + 3n + |T| + n_a + (1 + 3n)n_b = 2(n_a + 1 + |T| - n)$$

We show that G' can be partitioned into 2 connected subgraphs $G[S], G[S']$ where $|S| = |S'| = n'/2$ if and only if B contains a matching.

First suppose that B contains a matching M. Let $S = \{a\} \cup P_a \cup (B - M)$ (and S' be other vertices) where P_a is the path of length n_a which is connected to a. It is straightforward to check that $|S| = n'/2$ and that $G[S], G[S']$ are both connected.

Conversely if such an S exists we can assume $a \in S$. It follows that $P_a \subseteq S$. Now $|S - (P \cup \{a\})| = |T| - n < n_b$. Now if a vertex $v \in \{b\} \cup A$ is in S, then it implies that the path of length n_b which is connected to this vertex is also in S. However, $|S - (P \cup \{a\})| < n_b$ and it implies that none of the vertices in $\{b\} \cup A$ are in S. Thus $S - (P \cup \{a\}) \subseteq B$. Let $M = B - S$. Now $|M| = n$ and M must be a matching as $A \subseteq S'$ means that M "covers" A.

Problem 4. Let $a, b, c \in \mathbb{Z}$ such that $a \neq b$, $a \neq c$, and $b \neq c$. Prove that for not all-equal (a, a^3), (b, b^3) and (c, c^3) are collinear if and only if $a + b + c = 0$.

Solution. We need to show that the three points (a, a^3), (b, b^3) and (c, c^3) are collinear if and only if $a + b + c = 0$. Let us assume that the three points are collinear. Then the point (c, c^3) satisfies the equation of the line that passes through the points (a, a^3) and (b, b^3), i.e., we have:

$$
\begin{aligned}
&\frac{c-a}{b-a} = \frac{c^3-a^3}{b^3-a^3} \\
&\Rightarrow \frac{c-a}{b-a} = \frac{(c-a)(c^2+ac+a^2)}{(b-a)(b^2+ab+a^2)} \\
&\Rightarrow b^2+ab+a^2 = c^2+ac+a^2 \\
&\Rightarrow b^2-c^2 = -a(b-c) \\
&\Rightarrow b+c = -a \\
&\Rightarrow a+b+c = 0
\end{aligned}
$$

Assuming $a + b + c = 0$ and following the same chain of equations in the opposite direction, we get that the required points are collinear.

Problem 5. Let $f : \mathbb{N} \to \mathbb{R}$ be a function, with $f(n) \geq n$. Prove that $\mathsf{NSPACE}(f(n)) \subseteq \mathsf{SPACE}(f(n)^2)$.
Note that above proves $\mathsf{NPSPACE} = \mathsf{PSPACE}$.

Solution. We first show that given a directed graph of n vertices and two special vertices s and t in the graph, we can determine if t is reachable from s using only $O(\log^2 n)$ space. Consider the following algorithm: Let $\text{reach}(u, v, k)$ be a boolean function that is true ($= 1$) if and only if there is a path from vertex u to v of length $\leq k$. We need to evaluate $\text{reach}(s, t, n)$. Note that $\text{reach}(u, v, k) = \exists w$ s.t. $\text{reach}(u, w, \lceil k/2 \rceil) \wedge \text{reach}(w, v, \lfloor k/2 \rfloor)$. This formulation leads to an easy recursive algorithm. As the length of the path reduces by a factor of 2 at each step, the recursion depth is $O(\log n)$. At each level of the recursion, we only need to space for the "guessed" vertex w that requires $O(\log n)$ bits. This leads to a total space complexity of $O(\log^2 n)$. Hence, we can determine if t is reachable from s in an n-node directed graph in $O(\log^2 n)$ space.
Now, for any language $L \in \mathsf{NSPACE}(f(n))$, we can construct a directed graph with $O(2^{f(n)})$ vertices (one vertex for each configuration of the turing machine) such that for any input x, the graph has a path from the starting configuration on input x to an accepting configuration if and only if $x \in L$. Hence, determining connectivity is sufficient to determine if $x \in L$. Now, using the above algorithm, we can determine s-t connectivity in $\mathsf{SPACE}(\log^2(2^{f(n)})) = \mathsf{SPACE}(f(n)^2)$.

Problem 6. Give a sub-cubic reduction from Negative-Triangle to Median.

Solution. Let $(G = (V, E), w)$ be the given instance of Negative Triangle. Consider the directed case, the proof for the undirected case is similar. Create a weighted directed graph (G', w'). Graph G' contains five copies A, B, B', C, C' of V. With the usual notation, v_A is the copy of v in A and similarly for the other sets. Let $Q = \Theta(M)$ be a large enough integer. For any pair of nodes u, v, we add the edges $u_A v_B$ of weight $Q + w(uv)$, $u_A v_{B'}$, of weight $Q - w(uv)$, $u_A v_C$ of weight $2Q - w(uv)$, $u_A v_{C'}$, of weight $2Q + w(uv)$, and $u_B v_C$ of weight $Q + w(uv)$. In this construction, when $uv \notin E$ (including the special case $u = v$), we simply assume $w(uv) = 2M$. Furthermore, we add a dummy node r, and edges rv_A and $v_A r$ of weight $Q/4$ for any $v \in V$.

In this graph we compute the median value M^*, and output YES to the input instance of Negative triangle iff $M^* < Q/4 + (n-1)Q/2 + 6nQ$. The running time of the algorithm is $\tilde{O}(m + T(O(n), O(M))) = \tilde{O}(T(n, M))$. Let us show its correctness. Next $d(.)$ denotes distances in G'. Observe that the median node has to be in $A \cup \{r\}$ since the remaining nodes cannot reach r. Note that

$$\text{Med}(r) \geq nQ/4 + 2n(Q/4 + 2Q - 2M) + 2n(Q/4 + Q - M) > Q/4 + (n-1)Q/2 + 6nQ$$

On the other hand, for any node v_a,

$$\begin{aligned}
\text{Med}(v_A) &= d(v_A, r) + \sum_{u \in V} d(v_A, u_A) + \sum_{u \in V} (d(v_A, u_B) + d(v_A, u_{B'})) \\
&\quad + \sum_{u \in V} (d(v_A, u_C) + d(v_A, u_{C'})) \\
&= Q/4 + (n-1)Q/2 + \sum_{u \in V} (Q + w(vu) + Q - w(vu)) + \sum_{u \in V} (d(v_A, u_C) + 2Q + w(vu)) \\
&= Q/4 + (n-1)Q/2 + 2nQ + \sum_{u \in V} (d(v_A, u_C) + 2Q + w(vu)) \\
&\leq Q/4 + (n-1)Q/2 + 6nQ
\end{aligned}$$

Therefore the median is in A. In the last inequality we upper bounded $d(v_A, u_C)$ with $w'(v_A u_C) = 2Q - w(vu)$. Observe that a strict inequality holds if there exists a third node z_B such that $w'(v_A z_B) + w'(z_B u_C) < w'(v_A u_C)$. Note that this can happen only if $vu \in E$, since otherwise $w'(v_A u_C) = 2Q - 2M \leq w'(v_A z_B) + w'(z_B u_C)$. Note also that, if either $vz \notin E$ or $zu \notin E$, $w'(v_A z_B) + w'(z_B u_C) \geq 2Q + M \geq w'(v_A u_c)$. Therefore we can conclude that the strict inequality holds iff there exists a triangle $\{v, z, u)$ in G such that $Q + w(vz) + Q + w(zu) < 2Q - w(vu)$, i.e. a negative triangle. The claim follows.