Karl Bringmann and Marvin Künnemann Summer 2019

Fine-Grained Complexity Theory, Exercise Sheet 5

www.mpi-inf.mpg.de/departments/algorithms-complexity/teaching/summer19/fine-complexity/

Total Points: **40** + 2 bonus points Due: Tuesday, **June 25**, 2019

You are allowed to collaborate on the exercise sheets, but you have to write down a solution on your own, ***using your own words****. Please indicate the names of your collaborators for each exercise you solve. Further, cite all external sources that you use (books, websites, research papers, etc.).*

You need to collect at least 50% of all points on exercise sheets to be admitted to the exam.

Exercise ◯◯ — 2 bonus points

Read the lecture notes (of the last three lectures), identify as many typos and other mistakes as you can, and add them as a list to your solutions. You get one bonus point for at least one typo/mistake and 2 bonus points for at least five typos/mistakes.

Exercise 1 — **5 + 5** points

In the lecture we generalized **3SUM** to the following problem:

k–**SUM:** Given k sets $A_1, A_2, \ldots, A_k$ of n integers each, determine whether there are $a_1 \in A_1$, $a_2 \in A_2, \ldots, a_k \in A_k$ such that $a_1 + a_2 + \ldots + a_k = 0$.

a) Demonstrate an algorithm solving k-**SUM** that runs in time $O(n^{k/2} \cdot \log n)$ for *even* k, and runs in time $O(n^{(k+1)/2})$ for *odd* k.

b) Describe how to generalize the $O(n^2 \cdot \operatorname{poly}\log\log n/\sqrt{\log n})$ time algorithm for **3SUM** from the lecture to k–**SUM** for any odd k. Can you obtain a similar improvement for even k?

Exercise 2 — **10** points

Consider the following problem that can be solved in time $O(n^2 \cdot \log n)$:

$\mathbf{X} + \mathbf{Y}$ problem: Given two sets X and Y of n integers, determine whether the multi-set

$$X + Y := \{a + b \mid a \in X, \, b \in Y\}$$

contains n^2 distinct integers.

Show that if the $\mathbf{X} + \mathbf{Y}$ problem can be solved in time $O(n^{2-\varepsilon})$ for some $\varepsilon > 0$, then **3SUM** on sets of N integers can be solved in time $O(N^{2-\varepsilon'})$ for some $\varepsilon' > 0$.

Exercise 3 — 5 points

Consider the following variation of the **Subset Sum** problem:

Unbounded Subset Sum: Given a set of n distinct integers $X = \{1 \leq x_1 < \cdots < x_n\}$ and an integer t, determine whether there are non-negative integers $\alpha_1, \ldots, \alpha_n$ such that taking the i–th element α_i times sums up to t, that is $\sum_{i \in [n]} \alpha_i \cdot x_i = t$.

Demonstrate an algorithm for the **Unbounded Subset Sum** problem running in time $\tilde{O}(n + t)$.

Exercise 4 — 4 + 2 + 2 + 4 + 3 points

Consider the following problem:

Zero Weight 3–Star: Given a weighted 4–partite graph $G = (V_1 \cup V_2 \cup V_3 \cup V_4, E)$, where $|V_1| = |V_2| = |V_3| = |V_4| = n$, determine whether there are vertices $v_1 \in V_1, v_2 \in V_2, v_3 \in V_3$, and $v_4 \in V_4$ such that they form a star of weight 0, that is $w(v_1, v_2) + w(v_1, v_3) + w(v_1, v_4) = 0$.

In this exercise, we show that any $O(N^{2-\varepsilon})$ time algorithm for **3SUM** on sets of size N implies an $O(n^{3-\varepsilon'})$ time algorithm for **Zero Weight 3–Star**, *and vice versa* (under randomized reductions).

a) Show that if there is an algorithm solving **3SUM** on sets of size N in time $O(N^{2-\varepsilon})$ for some $\varepsilon > 0$, then there is an algorithm solving **Zero Weight 3–Star** running in time $O(n^{3-\varepsilon'})$ for some $\varepsilon' > 0$.

   *Hint: Try to find an algorithm for **Zero Weight 3–Star** running in time $O(n^3)$ first.*

In the remaining exercises, we now proceed to show the surprising direction of the equivalence, namely that **Zero Weight 3–Star** is **3SUM**–hard at cubic time.

b) Show that if there is an $O(n^{3-\varepsilon})$ time algorithm (for some $\varepsilon > 0$) for **Zero Weight 3–Star**, then there is an $O(n^{3-\varepsilon'})$ time algorithm (for some $\varepsilon' > 0$), that decides whether at least one of n given **3SUM** instances is a "YES" instance.

c) Show that if there is an $O(q \cdot N^{2-\varepsilon})$ time algorithm (for some $\varepsilon > 0$) deciding whether at least one of q given **3SUM** instances (of size N each) is a "YES" instance, then there is also an $O(q \cdot M^{2-\varepsilon'})$ time algorithm (for some $\varepsilon' > 0$) deciding whether at least one of q given **Convolution–3SUM** instances (of size M each) is a "YES" instance.

d) Fix $0 < \alpha < 1$ and let $t = t(n) = n^\alpha$. Show that if we can decide in time $O(M^{2-\varepsilon})$ (for some $\varepsilon > 0$) whether at least one of $(M/t)^2$ **Convolution–3SUM** instances of size t is a "YES" instance, then we can also solve a single **Convolution–3SUM** instance of size M in time $O(M^{2-\varepsilon'})$ (for some $\varepsilon' > 0$).

e) Combine the parts b) to d) with the reduction from **3SUM** to **Convolution–3SUM** to obtain the desired reduction from **3SUM** to **Zero Weight 3–Star**, than is show that an $O(n^{3-\varepsilon})$ time algorithm for **Zero Weight 3–Star** (for some $\varepsilon > 0$) would imply a randomized $O(N^{2-\varepsilon'})$ time algorithm for **3SUM** (of sets of size N and for some $\varepsilon' > 0$) with error probability $O(N^{-100})$.