Exercise X, Sublinear Algorithms for Big Data Analysis 2024-2025

These exercises are for your own benefit. Feel free to collaborate and share your answers with other students, and solve as many problems as you can. Problems marked (*) are more difficult, but also more rewarding. These problems have been taken from various sources on the Internet, too numerous to cite individually.

1 Give a single pass streaming algorithm that find a 2-approximation to the maximum matching in the input graph $G = (V, E)$ with n vertices using $O(n \log n)$ bits of space.

Solution. Any inclusion-wise maximal matching M is a factor 2 approximation to the maximum matching in G. Such a matching can be trivially maintained using $O(n \log n)$ bits of space. □

2 Let $A \subseteq [n]$ be an arithmetic-progression-free set, i.e. a set such that for every triple $a, b, c \in A$ if $2b = a + c$, then $a = b = c$. Let $G = (V, E)$, $V = [3n]$, be defined as follows. For every $i \in [n]$ let $M_i = \{(a + i, a + 2i) : a \in A\}$. Let $E = \bigcup_{i \in [n]} M_i$. Show that M_i are induced, i.e. for every i it holds that the subgraph induces by the vertices matched by M_i contains only the edges of M_i.

Solution. Suppose not. Then there exist $i, j \in [n]$ such that for distinct $a, b, c \in A$ one has

$$\begin{aligned} i + a &= j + c \\ 2i + b &= 2j + c \end{aligned}$$

This means that $j - i = a - c$, so

$$b = c + 2(j - i) = c + 2(a - c) = 2a - c,$$

i.e. $b + c = 2a$, and a, b, c form an arithmetic progression of length 3. □

3 Pick integers d, s and let t be the smallest integer such that $(2d + 1)^t \geq n$. Let $A_{d,s}$ be the set of integers of the form $\sum_{i=0}^{t} a_i (2d + 1)^i$ with integers a_i satisfying

1. For all i, $0 \leq a_i \leq d$
2. $\sum_{i=0}^{t} a_i^2 = s$.

3a For every d, s the set $A_{d,s}$ has no three-term arithmetic progressions.

3b For $d = \lceil 2^{\sqrt{\log n}} \rceil$ and some choice of s, $|A_{d,s}| \geq n / 2^{O(\sqrt{\log n})} = n^{1-o(1)}$.

Solution. Note that we are essentially writing our numbers base $2d + 1$. For every $a = (a_0, a_1, \ldots, a_t), b = (b_0, b_1, \ldots, b_t), c = (c_0, c_1, \ldots, c_t)$ of the form above we have $b = (a + c)/2$ as integers if and only if $b = (a + c)/2$ as vectors – this is because there is no carry in the addition $a + c$ due to the restriction that all digits in the base-$(2d + 1)$ representation are all between 0 and d. To see that there are no length three arithmetic progressions, it suffices to note that the average of two distinct vectors of equal Euclidean norm necessarily has strictly smaller norm. This establishes **2a**.

To establish **2b**, fist note that at least a $2^{-O(\sqrt{\log n})}$ fraction of numbers between 0 and $(2d+1)^{t+1}$ satisfy the property that all their digits base $2d+1$ are between 0 and d, and therefore at least a $2^{-O(\sqrt{\log n})}$ fraction of numbers between 0 and $n-1$ do as well. Furthermore, note that the union of $A_{d,s}$ over all integer s between 0 and td^2 contains the entire set of integers between 0 and n with digits base $2d + 1$ bounded by d, so for at least one value of s the corresponding set $A_{d,s}$ has size at least a $1/(td^2) = 1/2^{O(\sqrt{\log n})}$ fraction of that, as required. □