Efficient reductions and algorithms for Subset Product

Pranjal Dutta (CMI), Mahesh Sreekumar Rajasree (IITK)

CALDAM 2023

- Introduction

- Introduction
- Randomized $\tilde{O}(n + t^{o(1)})$ expected algorithm for Subset-Product

- Introduction
- Randomized $\tilde{O}(n + t^{o(1)})$ expected algorithm for Subset-Product
- Subset Product to (Simultaneous) Subset Sum

- Introduction

- Randomized $\tilde{O}(n + t^{o(1)})$ expected algorithm for Subset-Product

- Subset Product to (Simultaneous) Subset Sum

- Hardness of Simultaneous Subset Sum

- Introduction
- Randomized $\tilde{O}(n + t^{o(1)})$ expected algorithm for Subset-Product
- Subset Product to (Simultaneous) Subset Sum
- Hardness of Simultaneous Subset Sum
- Conclusion

Introduction

Introduction

Introduction

Subset sum problem (SSUM) - Given $a_1, \ldots, a_n, t \in Z_{\geq 0}$, decide whether there exist $S \subseteq [n]$ such that

Introduction

Subset sum problem (SSUM) - Given $a_1, \ldots, a_n, t \in Z_{\geq 0}$, decide whether there exist $S \subseteq [n]$ such that

$$\sum_{i \in S} a_i = t$$

Introduction

Subset sum problem (SSUM) - Given $a_1, \ldots, a_n, t \in Z_{\geq 0}$, decide whether there exist $S \subseteq [n]$ such that

$$\sum_{i \in S} a_i = t$$

NP complete problem.

Introduction

Subset sum problem (SSUM) - Given $a_1, \ldots, a_n, t \in Z_{\geq 0}$, decide whether there exist $S \subseteq [n]$ such that

$$\sum_{i \in S} a_i = t$$

NP complete problem.

$O(nt)$ time algorithm due to Bellman.

Introduction

Subset sum problem (SSUM) - Given $a_1, \ldots, a_n, t \in Z_{\geq 0}$, decide whether there exist $S \subseteq [n]$ such that

$$\sum_{i \in S} a_i = t$$

NP complete problem.

$O(nt)$ time algorithm due to Bellman.

Randomized $\tilde{O}(n + t)$ time algorithm due to [Jin & Wu,Bringmann].

Introduction

Introduction

Subset product problem (SPROD) - Given $a_1, \ldots, a_n, t \in Z_{\geq 0}$, decide whether there exist $S \subseteq [n]$ such that

Introduction

Subset product problem (SPROD) - Given $a_1, \ldots, a_n, t \in Z_{\geq 0}$, decide whether there exist $S \subseteq [n]$ such that

$$\prod_{i \in S} a_i = t$$

Introduction

Subset product problem (SPROD) - Given $a_1, \ldots, a_n, t \in Z_{\geq 0}$, decide whether there exist $S \subseteq [n]$ such that

$$\prod_{i \in S} a_i = t$$

NP-complete problem. $O(nt)$ time algorithm due to Bellman.

Introduction

Subset product problem (SPROD) - Given $a_1, \ldots, a_n, t \in Z_{\geq 0}$, decide whether there exist $S \subseteq [n]$ such that

$$\prod_{i \in S} a_i = t$$

NP-complete problem. $O(nt)$ time algorithm due to Bellman.

Randomized $\tilde{O}(n + t)$ time algorithm?

Introduction

Subset product problem (SPROD) - Given $a_1, \ldots, a_n, t \in Z_{\geq 0}$, decide whether there exist $S \subseteq [n]$ such that

$$\prod_{i \in S} a_i = t$$

NP-complete problem. $O(nt)$ time algorithm due to Bellman.

Randomized $\tilde{O}(n + t)$ time algorithm?

- Simply take log? But won't work :(

$\tilde{O}(n + t)$ algorithm for SSUM

$\tilde{O}(n + t)$ algorithm for SSUM

Given $a_1, \ldots, a_n, t \in Z_{\geq 0}$, decide whether there exist $S \subseteq [n]$ such that

$\tilde{O}(n + t)$ algorithm for SSUM

Given $a_1, \ldots, a_n, t \in Z_{\geq 0}$, decide whether there exist $S \subseteq [n]$ such that

$$\sum_{i \in S} a_i = t$$

$\tilde{O}(n + t)$ algorithm for SSUM

Given $a_1, \ldots, a_n, t \in Z_{\geq 0}$, decide whether there exist $S \subseteq [n]$ such that

$$\sum_{i \in S} a_i = t$$

Consider

$\tilde{O}(n+t)$ algorithm for SSUM

Given $a_1, \ldots, a_n, t \in Z_{\geq 0}$, decide whether there exist $S \subseteq [n]$ such that

$$\sum_{i \in S} a_i = t$$

Consider

$$f(x) = \prod_{i \in [n]} (1 + x^{a_i})$$

$\tilde{O}(n+t)$ algorithm for SSUM

Given $a_1, \ldots, a_n, t \in Z_{\geq 0}$, decide whether there exist $S \subseteq [n]$ such that

$$\sum_{i \in S} a_i = t$$

Consider

$$f(x) = \prod_{i \in [n]} (1 + x^{a_i})$$

Claim:- $(a_1, \ldots, a_n, t) \in SSUM \iff coeff(f, x^t) \neq 0$

$\tilde{O}(n+t)$ algorithm for SSUM

Given $a_1, \ldots, a_n, t \in Z_{\geq 0}$, decide whether there exist $S \subseteq [n]$ such that

$$\sum_{i \in S} a_i = t$$

Consider

$$f(x) = \prod_{i \in [n]} (1 + x^{a_i})$$

Claim:- $(a_1, \ldots, a_n, t) \in SSUM \iff coeff(f, x^t) \neq 0$

$$f(x) = 1 + x^{a_1} + \ldots + x^{a_n} + x^{a_1+a_2} + x^{a_1+a_3} + \ldots + x^{a_1+a_2+\ldots+a_n}$$

Randomized $\tilde{O}(n + t^{o(1)})$ time algorithm for SPROD

$\tilde{O}(n + t)$ algorithm for SPROD

$\tilde{O}(n + t)$ algorithm for SPROD

Given $a_1, \ldots, a_n, t \in Z_{\geq 0}$, decide whether there exist $S \subseteq [n]$ such that

$\tilde{O}(n + t)$ algorithm for SPROD

Given $a_1, \ldots, a_n, t \in Z_{\geq 0}$, decide whether there exist $S \subseteq [n]$ such that

$$\prod_{i \in S} a_i = t$$

$\tilde{O}(n + t)$ algorithm for SPROD

Given $a_1, \ldots, a_n, t \in Z_{\geq 0}$, decide whether there exist $S \subseteq [n]$ such that

$$\prod_{i \in S} a_i = t$$

Let $a_i = \prod_j p_j^{e_{ij}}$ and $t = \prod_j p_j^{t_j}$. Then,

$\tilde{O}(n + t)$ algorithm for SPROD

Given $a_1, \ldots, a_n, t \in Z_{\geq 0}$, decide whether there exist $S \subseteq [n]$ such that

$$\prod_{i \in S} a_i = t$$

Let $a_i = \prod_j p_j^{e_{ij}}$ and $t = \prod_j p_j^{t_j}$. Then,

$$\prod_{i \in S} a_i = t \iff \prod_{i \in S} \prod_j p_j^{e_{ij}} = \prod_j p_j^{t_j} \iff \prod_j p_j^{\sum_i e_{ij}} = \prod_j p_j^{t_j} \iff \sum_{i \in S} e_{ij} = t_j, \forall j$$

$\tilde{O}(n+t)$ algorithm for SPROD

Given $a_1, \ldots, a_n, t \in Z_{\geq 0}$, decide whether there exist $S \subseteq [n]$ such that

$$\prod_{i \in S} a_i = t$$

Let $a_i = \prod_j p_j^{e_{ij}}$ and $t = \prod_j p_j^{t_j}$. Then,

$$\prod_{i \in S} a_i = t \iff \prod_{i \in S} \prod_j p_j^{e_{ij}} = \prod_j p_j^{t_j} \iff \prod_j p_j^{\sum_i e_{ij}} = \prod_j p_j^{t_j} \iff \sum_{i \in S} e_{ij} = t_j, \forall j$$

Given k SSUM instances $e_{1j}, \ldots, e_{nj}, t_j \in Z_{\geq 0}$, decide whether there exists $S \subseteq [n]$ such that

$\tilde{O}(n + t)$ algorithm for SPROD

Given $a_1, \ldots, a_n, t \in Z_{\geq 0}$, decide whether there exist $S \subseteq [n]$ such that

$$\prod_{i \in S} a_i = t$$

Let $a_i = \prod_j p_j^{e_{ij}}$ and $t = \prod_j p_j^{t_j}$. Then,

$$\prod_{i \in S} a_i = t \iff \prod_{i \in S} \prod_j p_j^{e_{ij}} = \prod_j p_j^{t_j} \iff \prod_j p_j^{\sum_i e_{ij}} = \prod_j p_j^{t_j} \iff \sum_{i \in S} e_{ij} = t_j, \forall j$$

Given k SSUM instances $e_{1j}, \ldots, e_{nj}, t_j \in Z_{\geq 0}$, decide whether there exists $S \subseteq [n]$ such that

$$\sum_{i \in S} e_{ij} = t_j, \forall j$$

$\tilde{O}(n + t)$ algorithm for SPROD

$\tilde{O}(n + t)$ algorithm for SPROD

Given k SSUM instances $e_{1j}, \ldots, e_{nj}, t_j \in Z_{\geq 0}$, decide whether there exists $S \subseteq [n]$ such that

$\tilde{O}(n + t)$ algorithm for SPROD

Given k SSUM instances $e_{1j}, \ldots, e_{nj}, t_j \in Z_{\geq 0}$, decide whether there exists $S \subseteq [n]$ such that

$$\sum_{i \in S} e_{ij} = t_j, \forall j$$

$\tilde{O}(n+t)$ algorithm for SPROD

Given k SSUM instances $e_{1j}, \ldots, e_{nj}, t_j \in Z_{\geq 0}$, decide whether there exists $S \subseteq [n]$ such that

$$\sum_{i \in S} e_{ij} = t_j, \forall j$$

Consider

$\tilde{O}(n+t)$ algorithm for SPROD

Given k SSUM instances $e_{1j}, \ldots, e_{nj}, t_j \in Z_{\geq 0}$, decide whether there exists $S \subseteq [n]$ such that

$$\sum_{i \in S} e_{ij} = t_j, \forall j$$

Consider

$$f(x_1, \ldots, x_k) = \prod_{i \in [n]} (1 + \prod_{j \in [k]} x_j^{e_{ij}})$$

$\tilde{O}(n+t)$ algorithm for SPROD

Given k SSUM instances $e_{1j}, \ldots, e_{nj}, t_j \in Z_{\geq 0}$, decide whether there exists $S \subseteq [n]$ such that

$$\sum_{i \in S} e_{ij} = t_j, \forall j$$

Consider

$$f(x_1, \ldots, x_k) = \prod_{i \in [n]} (1 + \prod_{j \in [k]} x_j^{e_{ij}})$$

Theorem:- There is an $\tilde{O}(kn + \prod_j (2t_j + 1))$ **algorithm for SimulSSUM.**

$\tilde{O}(n + t)$ algorithm for SPROD

$\tilde{O}(n + t)$ algorithm for SPROD

Theorem:- There is an $\tilde{O}(kn + \prod_j (2t_j + 1))$ algorithm for SimulSSUM.

$\tilde{O}(n+t)$ algorithm for SPROD

Theorem:- There is an $\tilde{O}(kn + \prod_j (2t_j + 1))$ algorithm for SimulSSUM.

$$\prod_{i \in S} a_i = t \iff \prod_j p_j^{\sum_i e_{ij}} = \prod_j p_j^{t_j} \iff \sum_{i \in S} e_{ij} = t_j, \forall j$$

$\tilde{O}(n + t)$ algorithm for SPROD

Theorem:- There is an $\tilde{O}(kn + \prod_j (2t_j + 1))$ algorithm for SimulSSUM.

$$\prod_{i \in S} a_i = t \iff \prod_j p_j^{\sum_i e_{ij}} = \prod_j p_j^{t_j} \iff \sum_{i \in S} e_{ij} = t_j, \forall j$$

k is the number of prime factors in t. Therefore, $k = O(\log(t))$.

$\tilde{O}(n + t)$ algorithm for SPROD

Theorem:- There is an $\tilde{O}(kn + \prod_j (2t_j + 1))$ algorithm for SimulSSUM.

$$\prod_{i \in S} a_i = t \iff \prod_j p_j^{\sum_i e_{ij}} = \prod_j p_j^{t_j} \iff \sum_{i \in S} e_{ij} = t_j, \forall j$$

k is the number of prime factors in t. Therefore, $k = O(\log(t))$.

$t = \prod_j p_j^{t_j}$. Therefore, $t_j \le \log(t)$.

$\tilde{O}(n + t)$ algorithm for SPROD

Theorem:- There is an $\tilde{O}(kn + \prod_j (2t_j + 1))$ algorithm for SimulSSUM.

$$\prod_{i \in S} a_i = t \iff \prod_j p_j^{\sum_i e_{ij}} = \prod_j p_j^{t_j} \iff \sum_{i \in S} e_{ij} = t_j, \forall j$$

k is the number of prime factors in t. Therefore, $k = O(\log(t))$.

$t = \prod_j p_j^{t_j}$. Therefore, $t_j \leq \log(t)$.

Time to compute e_{ij}, t_j is $\tilde{O}(t)$. Solving SimulSSUM takes $\tilde{O}(n + t)$.

$\tilde{O}(n+t)$ algorithm for SPROD

Theorem:- There is an $\tilde{O}(kn + \prod_j (2t_j + 1))$ algorithm for SimulSSUM.

$$\prod_{i \in S} a_i = t \iff \prod_j p_j^{\sum_i e_{ij}} = \prod_j p_j^{t_j} \iff \sum_{i \in S} e_{ij} = t_j, \forall j$$

k is the number of prime factors in t. Therefore, $k = O(\log(t))$.

$t = \prod_j p_j^{t_j}$. Therefore, $t_j \leq \log(t)$.

Time to compute e_{ij}, t_j is $\tilde{O}(t)$. Solving SimulSSUM takes $\tilde{O}(n+t)$.

By considering $k = O(\log(t)/\log\log(t))$, we show $\tilde{O}(n + t^{o(1)})$.

Subset Product to Simultaneous Subset Sum

SPROD to SimulSSUM

SPROD to SimulSSUM

Given $a_1, \ldots, a_n, t \in Z_{\geq 0}$, decide whether there exist $S \subseteq [n]$ such that

SPROD to SimulSSUM

Given $a_1, \ldots, a_n, t \in Z_{\geq 0}$, decide whether there exist $S \subseteq [n]$ such that

$$\prod_{i \in S} a_i = t$$

SPROD to SimulSSUM

Given $a_1, \ldots, a_n, t \in Z_{\geq 0}$, decide whether there exist $S \subseteq [n]$ such that

$$\prod_{i \in S} a_i = t$$

Let $a_i = \prod_j p_j^{e_{ij}}$ and $t = \prod_j p_j^{t_j}$. Then,

SPROD to SimulSSUM

Given $a_1, \ldots, a_n, t \in Z_{\geq 0}$, decide whether there exist $S \subseteq [n]$ such that

$$\prod_{i \in S} a_i = t$$

Let $a_i = \prod_j p_j^{e_{ij}}$ and $t = \prod_j p_j^{t_j}$. Then,

$$\prod_{i \in S} a_i = t \iff \sum_{i \in S} e_{ij} = t_j, \forall j$$

SPROD to SimulSSUM

Given $a_1, \ldots, a_n, t \in Z_{\geq 0}$, decide whether there exist $S \subseteq [n]$ such that

$$\prod_{i \in S} a_i = t$$

Let $a_i = \prod_j p_j^{e_{ij}}$ and $t = \prod_j p_j^{t_j}$. Then,

$$\prod_{i \in S} a_i = t \iff \sum_{i \in S} e_{ij} = t_j, \forall j$$

Is it necessary that p_j's need to be prime?

SPROD to SimulSSUM

Given $a_1, \ldots, a_n, t \in Z_{\geq 0}$, decide whether there exist $S \subseteq [n]$ such that

$$\prod_{i \in S} a_i = t$$

Let $a_i = \prod_j p_j^{e_{ij}}$ and $t = \prod_j p_j^{t_j}$. Then,

$$\prod_{i \in S} a_i = t \iff \sum_{i \in S} e_{ij} = t_j, \forall j$$

Is it necessary that p_j's need to be prime?

No. Coprimality suffices!

SPROD to SimulSSUM

SPROD to SimulSSUM

Given $a_1, \ldots, a_n \in Z_{\geq 0}$, find $P \subset \mathbb{N}$ such that

SPROD to SimulSSUM

Given $a_1, \ldots, a_n \in Z_{\geq 0}$, find $P \subset \mathbb{N}$ such that

- Elements of P are pairwise coprime.

SPROD to SimulSSUM

Given $a_1, \ldots, a_n \in Z_{\geq 0}$, find $P \subset \mathbb{N}$ such that

- Elements of P are pairwise coprime.
- There is a non-trivial factor $(\neq 1)$ of some a_i in P.

SPROD to SimulSSUM

Given $a_1, \ldots, a_n \in Z_{\geq 0}$, find $P \subset \mathbb{N}$ such that

- Elements of P are pairwise coprime.
- There is a non-trivial factor ($\neq 1$) of some a_i in P.
- All a_i can be uniquely expressed in-terms of elements of P.

SPROD to SimulSSUM

Given $a_1, \ldots, a_n \in Z_{\geq 0}$, find $P \subset \mathbb{N}$ such that

- Elements of P are pairwise coprime.
- There is a non-trivial factor ($\neq 1$) of some a_i in P.
- All a_i can be uniquely expressed in-terms of elements of P.

For example. Consider (6, 30, 77). Then a possible P is (6, 5, 77).

SPROD to SimulSSUM

Given $a_1, \ldots, a_n \in Z_{\geq 0}$, find $P \subset \mathbb{N}$ such that

- Elements of P are pairwise coprime.
- There is a non-trivial factor ($\neq 1$) of some a_i in P.
- All a_i can be uniquely expressed in-terms of elements of P.

For example. Consider (6, 30, 77). Then a possible P is (6, 5, 77).

- 6 = gcd(6, 30) and is coprime to 77

SPROD to SimulSSUM

Given $a_1, \ldots, a_n \in Z_{\geq 0}$, find $P \subset \mathbb{N}$ such that

- Elements of P are pairwise coprime.
- There is a non-trivial factor ($\neq 1$) of some a_i in P.
- All a_i can be uniquely expressed in-terms of elements of P.

For example. Consider (6, 30, 77). Then a possible P is (6, 5, 77).

- 6 = gcd(6, 30) and is coprime to 77
- Since 6 | 30, we consider 5 and our list reduces to (5,77)

SPROD to SimulSSUM

Given $a_1, \ldots, a_n \in Z_{\geq 0}$, find $P \subset \mathbb{N}$ such that

- Elements of P are pairwise coprime.
- There is a non-trivial factor ($\neq 1$) of some a_i in P.
- All a_i can be uniquely expressed in-terms of elements of P.

For example. Consider (6, 30, 77). Then a possible P is (6, 5, 77).

- 6 = gcd(6, 30) and is coprime to 77
- Since 6 | 30, we consider 5 and our list reduces to (5,77)
- Since 5 and 77 are coprime, return as it is.

Pseudo-Prime Factors

Pseudo-Prime Factors

Given $a_1, \ldots, a_n \in Z_{\geq 0}$, find $P \subset \mathbb{N}$ such that

Pseudo-Prime Factors

Given $a_1, \ldots, a_n \in Z_{\geq 0}$, find $P \subset \mathbb{N}$ such that

- Elements of P are pairwise coprime.

Pseudo-Prime Factors

Given $a_1, \ldots, a_n \in Z_{\geq 0}$, find $P \subset \mathbb{N}$ such that

- Elements of P are pairwise coprime.
- There is a non-trivial factor of some a_i in P.

Pseudo-Prime Factors

Given $a_1, \ldots, a_n \in Z_{\geq 0}$, find $P \subset \mathbb{N}$ such that

- Elements of P are pairwise coprime.
- There is a non-trivial factor of some a_i in P.
- All a_i can be uniquely expressed in-terms of elements of P.

Pseudo-Prime Factors

Given $a_1, \ldots, a_n \in Z_{\geq 0}$, find $P \subset \mathbb{N}$ such that

- Elements of P are pairwise coprime.
- There is a non-trivial factor of some a_i in P.
- All a_i can be uniquely expressed in-terms of elements of P.

Lemma 1: If a_1 is co-prime to $a_2, a_3, \ldots, a_n$ and P' is a pseudo-prime factor set of $(a_2, \ldots, a_n)$, then $\{a_1\} \cup P'$ is a pseudo-prime factor set of $(a_1, \ldots, a_n)$.

Pseudo-Prime Factors

Given $a_1, \ldots, a_n \in Z_{\geq 0}$, find $P \subset \mathbb{N}$ such that

- Elements of P are pairwise coprime.
- There is a non-trivial factor of some a_i in P.
- All a_i can be uniquely expressed in-terms of elements of P.

Lemma 1: If a_1 is co-prime to $a_2, a_3, \ldots, a_n$ and P' is a pseudo-prime factor set of $(a_2, \ldots, a_n)$, then $\{a_1\} \cup P'$ is a pseudo-prime factor set of $(a_1, \ldots, a_n)$.

Lemma 2: Let g be a factor of some prime a_i. Set $b_j = a_j // g$, i.e., b_j is not divisible by g. Then, a pseudo-prime factor set of $(g, b_1, \ldots, b_n)$ is also the same for $(a_1, \ldots, a_n)$.

Pseudo-Prime Factors

Pseudo-Prime Factors

Given $a_1, \ldots, a_n \in Z_{\geq 0}$, find a pseudo-prime factor set $P \subset \mathbb{N}$ for $(a_1, \ldots, a_n)$.

Pseudo-Prime Factors

Given $a_1, \ldots, a_n \in Z_{\geq 0}$, find a pseudo-prime factor set $P \subset \mathbb{N}$ for $(a_1, \ldots, a_n)$.

Lemma 1: If a_1 is co-prime to $a_2, a_3, \ldots, a_n$ and P' is a pseudo-prime factor set of $(a_2, \ldots, a_n)$, then $\{a_1\} \cup P'$ is a pseudo-prime factor set of $(a_1, \ldots, a_n)$.

Pseudo-Prime Factors

Given $a_1, \ldots, a_n \in Z_{\geq 0}$, find a pseudo-prime factor set $P \subset \mathbb{N}$ for $(a_1, \ldots, a_n)$.

Lemma 1: If a_1 is co-prime to $a_2, a_3, \ldots, a_n$ and P' is a pseudo-prime factor set of $(a_2, \ldots, a_n)$, then $\{a_1\} \cup P'$ is a pseudo-prime factor set of $(a_1, \ldots, a_n)$.

Lemma 2: Let g be a factor of some prime a_i. Set $b_i = a_i // g$, i.e., b_i is not divisible by g. Then, a pseudo-prime factor set of $(g, b_1, \ldots, b_n)$ is also the same for $(a_1, \ldots, a_n)$.

Pseudo-Prime Factors

Given $a_1, \ldots, a_n \in Z_{\geq 0}$, find a pseudo-prime factor set $P \subset \mathbb{N}$ for $(a_1, \ldots, a_n)$.

Lemma 1: If a_1 is co-prime to $a_2, a_3, \ldots, a_n$ and P' is a pseudo-prime factor set of $(a_2, \ldots, a_n)$, then $\{a_1\} \cup P'$ is a pseudo-prime factor set of $(a_1, \ldots, a_n)$.

Lemma 2: Let g be a factor of some prime a_i. Set $b_i = a_i // g$, i.e., b_i is not divisible by g. Then, a pseudo-prime factor set of $(g, b_1, \ldots, b_n)$ is also the same for $(a_1, \ldots, a_n)$.

In lemma 1, the size of the set decreases by 1.

Pseudo-Prime Factors

Given $a_1, \ldots, a_n \in Z_{\geq 0}$, find a pseudo-prime factor set $P \subset \mathbb{N}$ for $(a_1, \ldots, a_n)$.

Lemma 1: If a_1 is co-prime to $a_2, a_3, \ldots, a_n$ and P' is a pseudo-prime factor set of $(a_2, \ldots, a_n)$, then $\{a_1\} \cup P'$ is a pseudo-prime factor set of $(a_1, \ldots, a_n)$.

Lemma 2: Let g be a factor of some prime a_i. Set $b_i = a_i // g$, i.e., b_i is not divisible by g. Then, a pseudo-prime factor set of $(g, b_1, \ldots, b_n)$ is also the same for $(a_1, \ldots, a_n)$.

In lemma 1, the size of the set decreases by 1.

In lemma 2, let $g = gcd(a_1, a_2)$. Then, $2 \leq g \leq a_1/2$. Worst case scenario is $(2, a_1/2, a_2/2, a_3, \ldots, a_n)$.

Hardness of Simultaneous Subset Sum

SimulSSUM $\leq_p$ SSUM

SimulSSUM $\leq_p$SSUM

Suppose we are given a SimulSSUM with $k = 2$, i.e.,

SimulSSUM $\leq_p$ SSUM

Suppose we are given a SimulSSUM with $k = 2$, i.e.,

$$(a_1, \ldots, a_n, t)$$

SimulSSUM $\leq_p$SSUM

Suppose we are given a SimulSSUM with $k = 2$, i.e.,

$$(a_1, \ldots, a_n, t)$$

$$(b_1, \ldots, b_n, s)$$

SimulSSUM $\leq_p$SSUM

Suppose we are given a SimulSSUM with $k = 2$, i.e.,

$$(a_1, \ldots, a_n, t)$$

$$(b_1, \ldots, b_n, s)$$

Let λ be a large constant. Then,

SimulSSUM $\leq_p$SSUM

Suppose we are given a SimulSSUM with $k = 2$, i.e.,

$$(a_1, \ldots, a_n, t)$$

$$(b_1, \ldots, b_n, s)$$

Let λ be a large constant. Then,

$$(\lambda a_1 + b_1, \quad \lambda a_2 + b_2, \quad \ldots, \quad \lambda a_n + b_n, \quad \lambda t + s)$$

SimulSSUM $\leq_p$SSUM

Suppose we are given a SimulSSUM with $k = 2$, i.e.,

$$(a_1, \ldots, a_n, t)$$

$$(b_1, \ldots, b_n, s)$$

Let λ be a large constant. Then,

$$(\lambda a_1 + b_1, \quad \lambda a_2 + b_2, \quad \ldots, \quad \lambda a_n + b_n, \quad \lambda t + s)$$

If $\sum_{i \in S} a_i = t, \sum_{i \in S} b_i = s \implies \sum_{i \in S} \lambda a_i + b_1 = \lambda t + s.$

SSUM $\leq_p$ SimulSSUM

SSUM $\leq_p$ SimulSSUM

Suppose we are given a SSUM, i.e.,

SSUM $\leq_p$ SimulSSUM

Suppose we are given a SSUM, i.e.,

$$(a_1, \ldots, a_n, t)$$

SSUM$\leq_p$SimulSSUM

Suppose we are given a SSUM, i.e.,

$$(a_1, \ldots, a_n, t)$$

Consider the SimulSSUM instance

SSUM $\leq_p$ SimulSSUM

Suppose we are given a SSUM, i.e.,

$$(a_1, \ldots, a_n, t)$$

Consider the SimulSSUM instance

$$(a_1, a_2, \ldots, a_n, t)$$

SSUM $\leq_p$ SimulSSUM

Suppose we are given a SSUM, i.e.,

$$(a_1, \ldots, a_n, t)$$

Consider the SimulSSUM instance

$$(a_1, a_2, \ldots, a_n, t)$$

$$(1, \quad \underbrace{0, \ldots, 0,}_{n-1} \quad b)$$

SSUM$\leq_p$SimulSSUM

Suppose we are given a SSUM, i.e.,

$$(a_1, \ldots, a_n, t)$$

Consider the SimulSSUM instance

$$(a_1, a_2, \ldots, a_n, t)$$

$$(1, \quad \underbrace{0,\ldots,0,}_{n-1} \quad b)$$

Claim:- If SSUM is YES, then SimulSSUM is YES for either $b = 0$ or $b = 1$.

SSUM$\leq_p$SimulSSUM

Suppose we are given a SSUM, i.e.,

$$(a_1, \ldots, a_n, t)$$

Consider the SimulSSUM instance

$$(a_1, a_2, \ldots, a_n, t)$$

$$(1, \underbrace{0,\ldots,0,}_{n-1} b)$$

Claim:- If SSUM is YES, then SimulSSUM is YES for either $b = 0$ or $b = 1$.

Claim:-If SSUM is NO, then SimulSSUM is NO for both $b = 0$ and $b = 1$.

Conclusion

Conclusion

- We saw an $\tilde{O}(n + t^{o(1)})$ time algorithm for SPROD.

Conclusion

- We saw an $\tilde{O}(n + t^{o(1)})$ time algorithm for SPROD.
- We saw an $\tilde{O}(kn + \prod_j (2t_j + 1))$ time algorithm for SimulSSUM.

Conclusion

- We saw an $\tilde{O}(n + t^{o(1)})$ time algorithm for SPROD.
- We saw an $\tilde{O}(kn + \prod_j (2t_j + 1))$ time algorithm for SimulSSUM.
- Polynomial time reduction from SPROD to SimulSSUM and SimulSSUM to SSUM

Conclusion

- We saw an $\tilde{O}(n + t^{o(1)})$ time algorithm for SPROD.
- We saw an $\tilde{O}(kn + \prod_j (2t_j + 1))$ time algorithm for SimulSSUM.
- Polynomial time reduction from SPROD to SimulSSUM and SimulSSUM to SSUM
- Can we improve the time complexity for SimulSSUM to $\tilde{O}(kn + \prod_j t_j)$?

Conclusion

- We saw an $\tilde{O}(n + t^{o(1)})$ time algorithm for SPROD.
- We saw an $\tilde{O}(kn + \prod_j (2t_j + 1))$ time algorithm for SimulSSUM.
- Polynomial time reduction from SPROD to SimulSSUM and SimulSSUM to SSUM
- Can we improve the time complexity for SimulSSUM to $\tilde{O}(kn + \prod_j t_j)$?
- Hardness of SimulSSUM for $k = \omega(\log(n))$?

Thank You!