Faster Space-Efficient Algorithms for Subset Sum

Nikhil Bansal[1], Shashwat Garg[1], Jesper Nederlof[1], Nikhil Vyas[2]

***1.** Eindhoven University of Technology, Eindhoven, Netherlands* **2.** Indian Institute of Technology Bombay, India

Abstract

We present randomized algorithms that solve Subset Sum and Knapsack Instances with n items in $O^*(2^{0.86n})$ time and polynomial space, assuming random read-only access to exponentially many random bits. Here $O^*()$ omits factor polynomial in the input size.

Underlying these results is an algorithm that determines whether two given lists of length n with integers bounded by a polynomial in n share a common value. Assuming random read-only access to random bits, we show that this problem can be solved using $O(\log n)$ space significantly faster than the trivial $O(n^2)$ time algorithm if no value occurs too often in the same list.

Introduction

Subset Sum (SSS)

$w(X) := \sum_{i \in X} w_i$

Given integers $w_1, \dots, w_n, t$,
is there $X \subseteq [n]$ with $w(X) = t$?

Example SSS instance

$w_1, \dots, w_{12} = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12\}$, $t = 50$

Main question: How efficiently can we solve SSS exactly?

Some known results

Much exciting recent progress!!

Instances with small t:

Classic DP by [Bellman (50's)] : $O^*(t)$ time and space

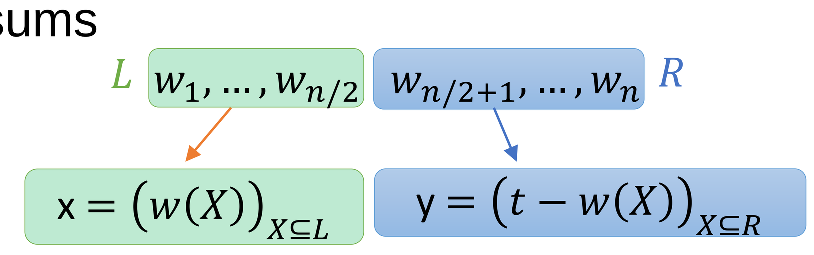
– Improved to $O^*(t)$ time and poly space [LN(STOC'10)]

Instance with large t (i.e. $t = 2^n$):

Meet-in-the-middle (MitM) [HS(JACM'74)]: $O^*(2^{n/2})$ time

1. Let $L = (w_1, \dots, w_{n/2})$ $R = (w_{n/2+1}, \dots, w_n)$
2. Compute all possible $2^{n/2}$ sums
3. For each $v \in x$,
   check $v \in y$

Sort y + binary search in 3:
$O^*(2^{n/2})$ time and space



– In $O^*(2^{n/2})$ time, $O^*(2^{n/4})$ space [SS(SICOMP'81)]

See right side poster

– In $O^*(2^{0.49991n})$ time, if $\geq 2^{0.997n}$ *distinct sums* [AKKN(STACS'16)]

Natural question: Can we beat $O^*(2^n)$ using *polynomial space*?

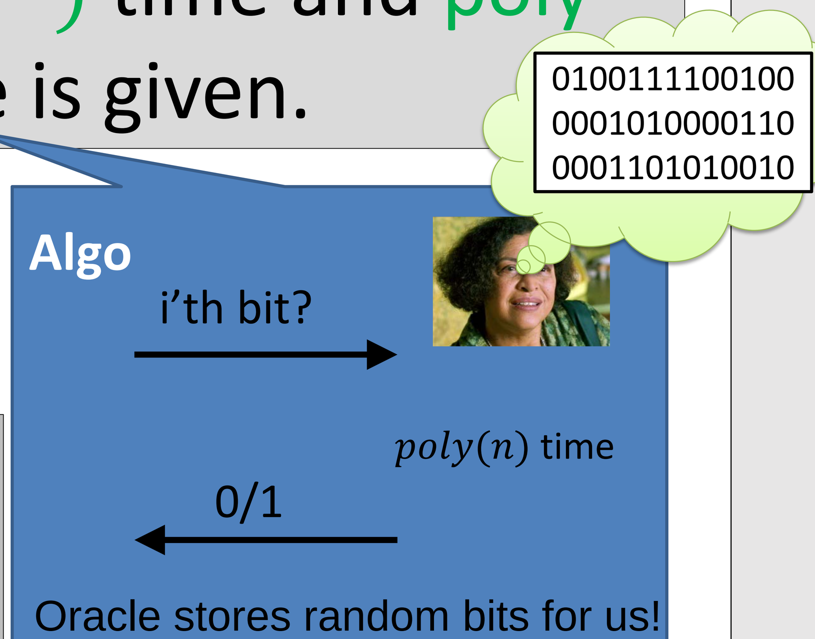
Our Contribution

Main theorem: SSS in $O^*(2^{0.86n})$ time and poly space, if given a random oracle is given.

- Weaker assumption than sufficiently strong PRG's

Main theorem generalizes to

Thm: Binary LP on n vars, $o(n/\log^2 n)$ constraints in time $O^*(2^{0.86n})$ and poly space, if given a random oracle



- Follows by combining Main theorem with [NLZ (MFCS'12)]

Also, random* 3SUM is solved in $O(n^{2.5})$ time and $O(\log n)$ space

*see the paper

Proof idea main theorem

Let $w(2^{[n]}) = \{w \cdot x : x \in \{0,1\}^n\}$
i.e. all possible 2^n sums generated by $w = (w_1, \dots, w_n)$
Let $d = |w(2^{[n]})|$ i.e. # distinct sums

Case 1 $d \leq 2^{0.86n}$ *(few distinct sums)*

a) Hash mod $O(d)$, which makes $t = O^*(d)$

By a union bound over all sums from $w(2^{[n]})$, introduce false positives with only constant probability

b) Use $O^*(t)$ time ***poly*** space algo [LN(STOC'10)]

Interpolates the polynomial $p(x) = \prod_{i=1}^{n}(1 + x^{w_i})$ to determine the coefficient of x^t using inverse DFT

Case 2 $d > 2^{0.86n}$ (many distinct sums)

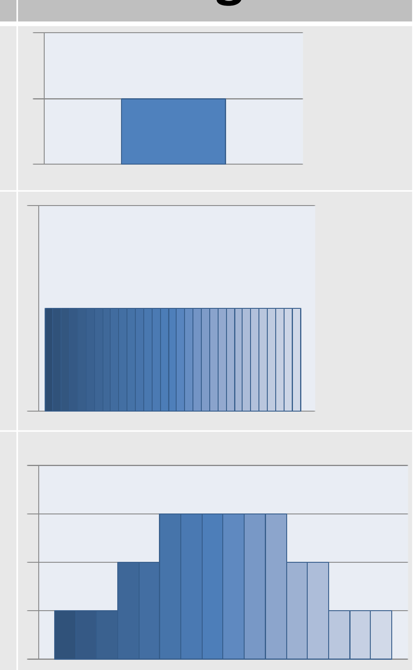
a) Upper bound *max bin size*

$$b_{max} = max_v |\{x \in \{0,1\}^l : w \cdot x = v\}|$$

Lemma AKKN(STACS'16)]: $d \cdot b_{max} \leq 2^{1.5n}$

- Subset Sum distribution smooth: cool AC result!
- Proved via simple connection to \`Uniquely Decodable Code Pairs'
- As $d > 2^{0.86n}$, we obtain $b_{max} \leq 2^{0.64n}$

w	d	b_{max}	Histogram
0 0 0 0 0	1	32	
1 2 4 8 16	32	1	
1 2 3 4 5	16	3	



b) Use Floyd's cycle finding

Idea: Use MitM without sorting. Need to solve problem similar to

Element Distinctness (ED)

Given list z of m ints, **find** two positions with equal ints, if exist

Ex. ED instance

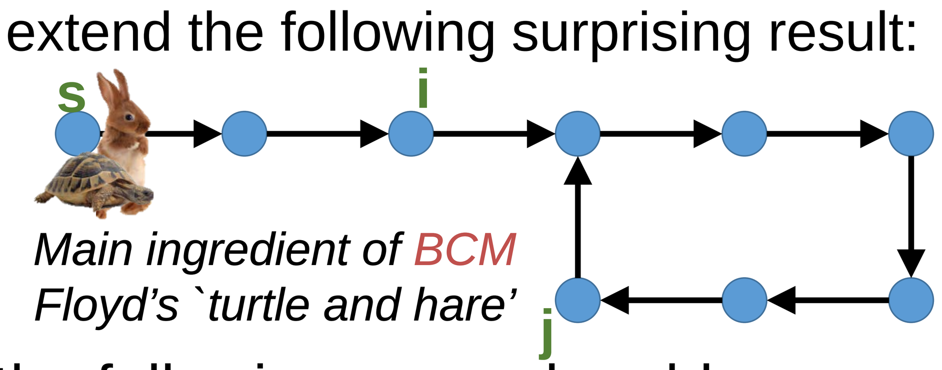
5 **3** 4 **3** 2 10 7 8 1 6

Repeat:
1. Define z as the concatenation of x and y
2. (Almost) uniformly sample a solution of ED instance z
3. Check if \`fake' solution or a real SSS solution

Crux: $O(\#fake\ sols) \leq O(2^{0.89n})$ by max bin bound!

How to sample fast in step 2.? We extend the following surprising result:

Thm [BCM(FOCS'13)]: ED in $\tilde{O}(m^{1.5})$ time, $O(\log m)$ space, if given random oracle



Main ingredient of BCM Floyd's \`turtle and hare'

And obtain an algorithm for the following general problem

List Disjointness (LD)

Given lists x, y, **find** two indices i, j s.t. x_i *equals* y_j.

Ex. LD instance

x 5 **3** 5 8 8 7   4 **3** 6 2 1 6 y

Denoting $f(x, y) = \sum_{v=1}^{m} |x^{-1}(v)|^2 + |y^{-1}(v)|^2$ for #fake sols, we get

Thm: LD In $\tilde{O}\left(m\sqrt{f}\right)$ time and $O(\log m)$ space, if given $f \geq f(x, y)$ and random oracle

Bibliography [AKKN(STACS'16)] Austrin, Kaski, Koivisto, Nederlof, *Dense Subset Sum may be the hardest* [BCM(FOCS'13)] Beame, Clifford, Machmouchi, *Element Distinctness, Frequency Moments, and Sliding Windows* [NLZ (MFCS'12)] Nederlof, van Leeuwen, van der Zwaan, *Reducing a Target Interval to a Few Exact Queries* [SS(SICOMP'81)] Schroeppel, Schamir *A T=O(2^{n/2}), S=O(2^{n/4}) Algorithm for Certain NP-Complete Problems* [HS(JACM'74)] Horowitz, Sahni *Computing Partitions with Applications to the Knapsack Problem* [LN(STOC'10)] Lokshtanov, Nederlof *Saving space by algebraization*