Computational Aspects of Ordered Integer Partition with Upper Bounds

Roland Glück[1] Dominik Köppl[1] Günther Wirsching[2]

[1]Universität Augsburg

[2]Katholische Universität Eichstätt

6.6.2013
Roma

Different Kinds of Partition

- (integer) partition: write $n \in \mathbb{N}^+$ as sum of positive integers
- partition number $p(n)$
- $4 = 4$, $4 = 3 + 1$, $4 = 2 + 2$, $4 = 2 + 1 + 1$, $4 = 1 + 1 + 1 + 1$
- $p(4) = 5$
- $p(n) \approx \frac{e^{(\pi\sqrt{\frac{2n}{3}})}}{4n\sqrt{3}}$
- also upper/lower bounds for summands

Different Kinds of Partition

- ordered partition (composition): write n as $n = \sum_{j=1}^{k} i_j$ with $i_j \in \mathbb{N}_{>0}$ for $1 \leq j \leq k$
- composition number $g(k, n)$
- $4 = 1 + 3$, $4 = 2 + 2$, $4 = 3 + 1$
- $g(2, 4) = 3$
- $g(k, n) = \binom{n-1}{k-1}$

Ordered Integer Partition with Upper Bounds

Definition

Given a sequence $I = (i_1, i_2, \ldots, i_n) \in \mathbb{N}_{>0}^n$ and a number $z \in \mathbb{Z}$. A sequence $Z = (z_1, z_2, \ldots, z_n) \in \mathbb{N}_0^n$ is called an *ordered integer partition with upper bounds of* z *wrt.* I (or ‘partition’ for short) if

- $\sum_{k=1}^{n} z_k = z$ and
- $0 \leq z_k \leq i_k$ for $1 \leq k \leq n$.

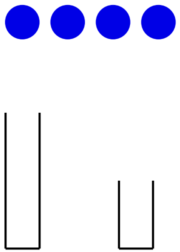
$\#(I, z)$ denotes the number of partitions of z wrt. I.
I is called *upper bounds*, z the *target value*.

Task: given I and z, compute $\#(I, z)$

Recent Work

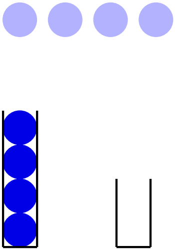
- problem arises in context of database preferences (Preference SQL) [Kießling, Endres, Preisinger et al.]
- combinatorial formulation: distribution of z balls into n urns with capacities $i_1, i_2, \ldots, i_n$
- considered by Wirsching for $n \to \infty$

Example



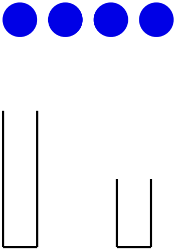
Example

$$4 = 4 + 0$$

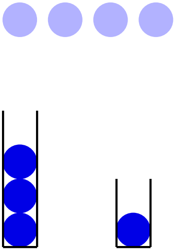


Example

$$4 = 4 + 0$$



Example



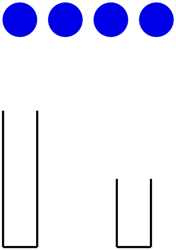
$$4 = 4 + 0$$

$$4 = 3 + 1$$

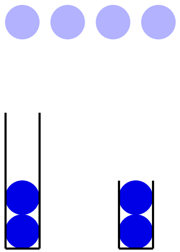
Example

$$4 = 4 + 0$$

$$4 = 3 + 1$$



Example

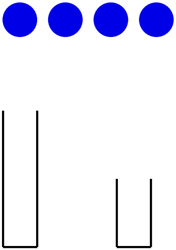


$$4 = 4 + 0$$

$$4 = 3 + 1$$

$$4 = 2 + 2$$

Example



$$4 = 4 + 0$$

$$4 = 3 + 1$$

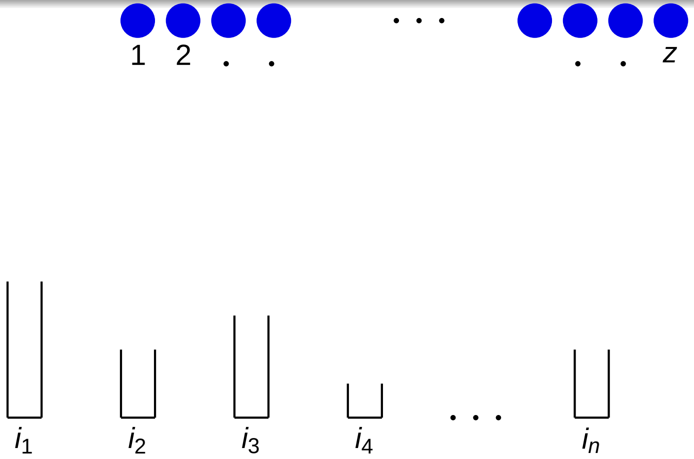
$$4 = 2 + 2$$

$$\#((4,2),4) = 3$$

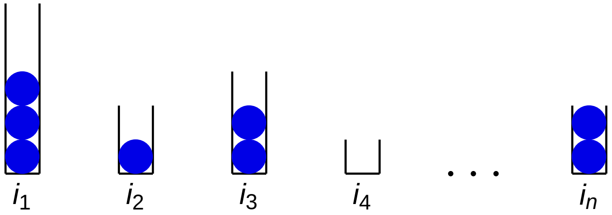
Simple Properties

- $\#((i_1), z) = \begin{cases} 1 & \text{if } 0 \leq z \leq i_1 \\ 0 & \text{otherwise} \end{cases}$
- $\#(I, z) = 0 \text{ if } z < 0$
- $\#((i_1, i_2, \ldots, i_n), z) = 0 \text{ if } z > \sum_{k=1}^{n} i_k := \sum I$

Symmetry Property

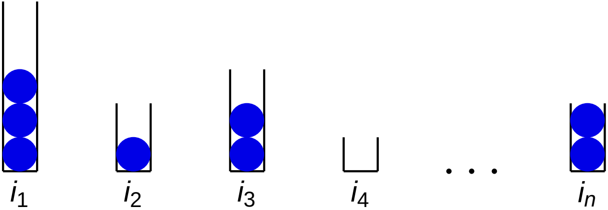


Symmetry Property



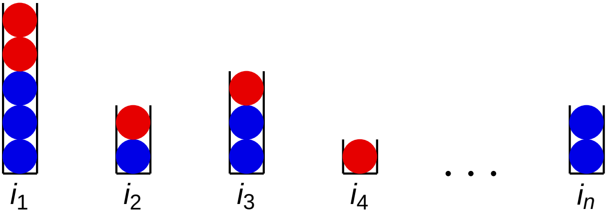
Symmetry Property

$$z_1 + z_2 + z_3 + z_4 + \cdots + z_n = z$$



Symmetry Property

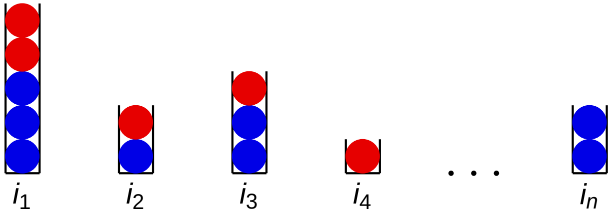
$$z_1 + z_2 + z_3 + z_4 + \cdots + z_n = z$$



Symmetry Property

$$i_1 - z_1 + i_2 - z_2 + i_3 - z_3 + i_4 - z_4 + \cdots + i_n - z_n = \sum I - z$$

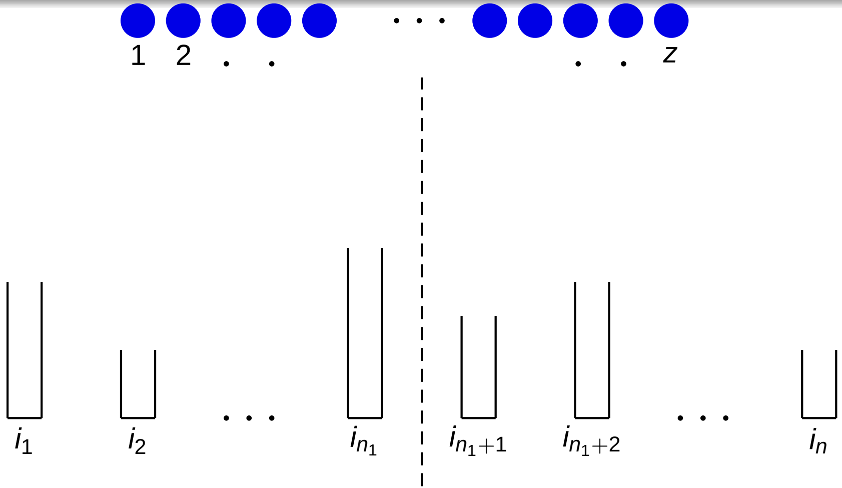
$$z_1 + z_2 + z_3 + z_4 + \cdots + z_n = z$$



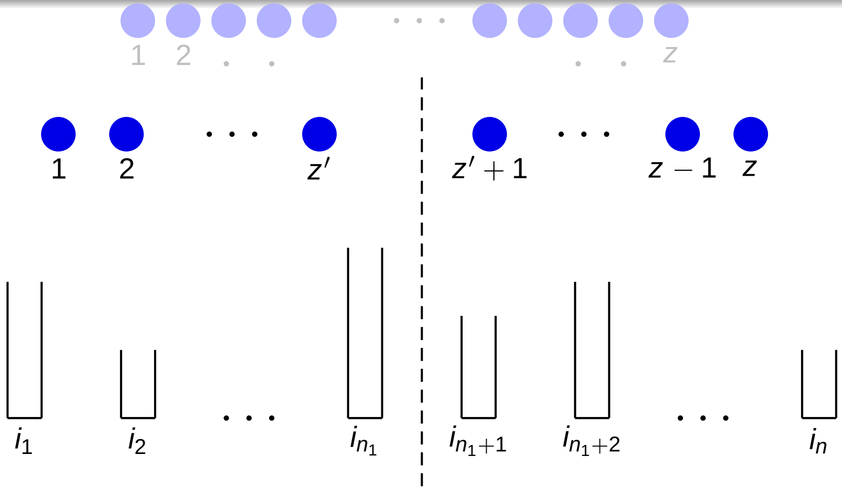
Symmetry Property

- bijection between partitions of z and $\sum I - z$
- hence $\#(I, z) = \#(I, \sum I - z)$
- symmetry of $\#(I, z)$ wrt. $\frac{\sum I}{2}$

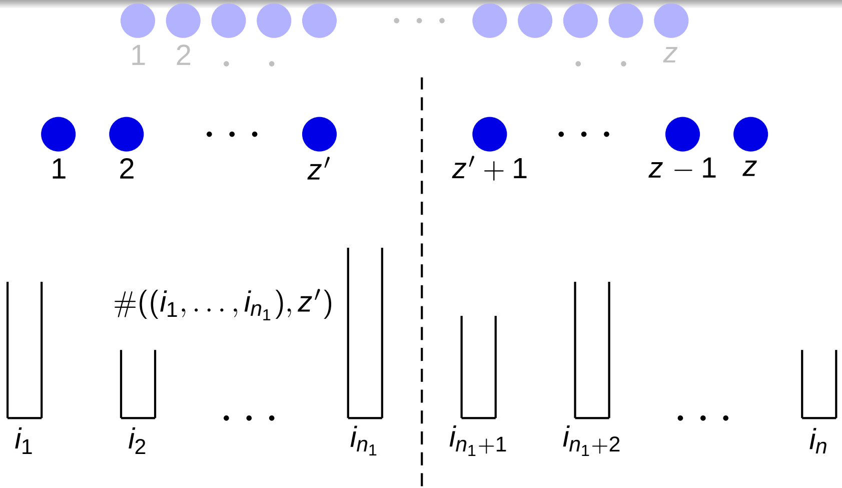
Splitting Property



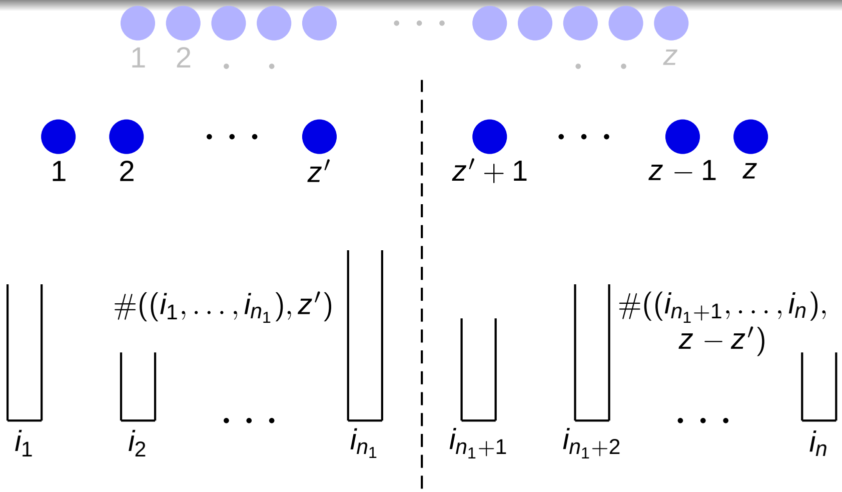
Splitting Property



Splitting Property

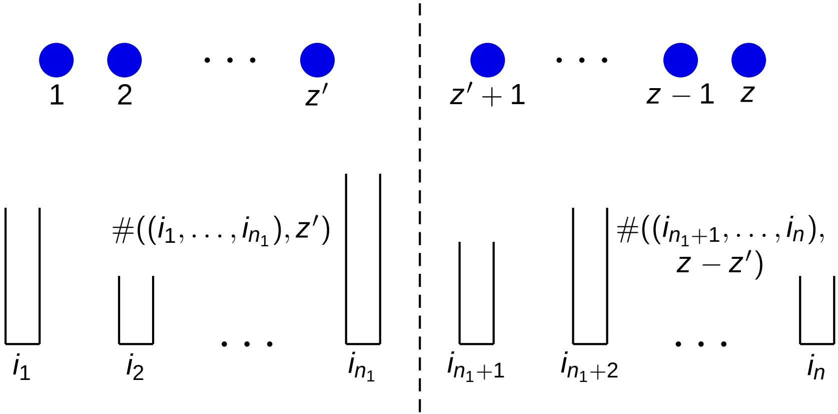


Splitting Property



Splitting Property

$$\#((i_1, \ldots, i_{n_1}), z') \cdot \#((i_{n_1+1}, \ldots, i_n), z - z')$$



Splitting Property

- $0 \leq z' \leq z$, so we have:

Theorem

$$\#((i_1, \ldots, i_n), z) = \sum_{z'=0}^{z} \#((i_1, \ldots, i_{n_1}), z') \cdot \#((i_{n_1+1}, \ldots, i_n), z - z'),$$ *in particular*

$$\#((i_1, \ldots, i_n), z) = \sum_{z'=0}^{z} \#((i_1, \ldots, i_{n-1}), z') \cdot \#((i_n), z - z')$$

Naïve Algorithm

shortcut for $\#((i_1, i_2), z)$ in O(1) leads to:

```
int part_num(upper_bounds I, int z){
  if(|I| <= 2) {return #(I, z);}
  if(z > ∑I/2) {z = ∑ I - z;}
  split I into I1 and I2;
  return ∑_{z'=0}^{z} part_num(I1,z')·part_num(I2,z-z');
}
```

Naïve Algorithm

shortcut for $\#((i_1, i_2), z)$ in O(1) leads to:

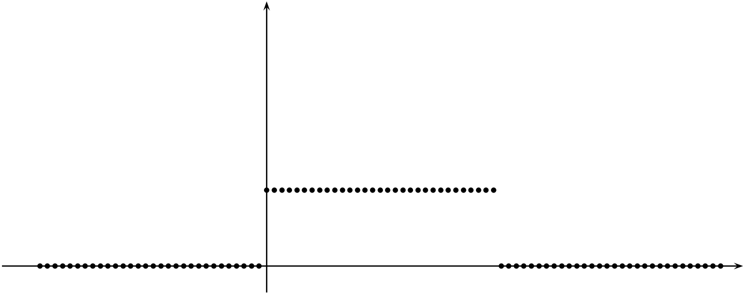
```
int part_num(upper_bounds I, int z){
  if(|I| <= 2) {return #(I, z);}
  if(z > ∑I/2) {z = ∑I - z;}
  split I into I1 and I2;
  return ∑_{z'=0}^{z} part_num(I1,z')·part_num(I2,z-z');
}
```

simple but unsatisfactory

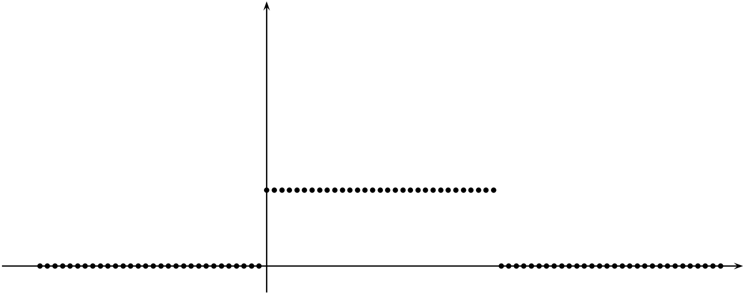
Towards a New Algorithm

- recall
  $$\#((i_1, \ldots, i_n), z) = \sum_{z'=0}^{z} \#((i_1, \ldots, i_{n-1}), z') \cdot \#((i_n), z - z')$$
- using properties of $\#((i_n), x)$ and index shifts lead to
  $$\#((i_1, \ldots, i_n), z) = \sum_{z'=z-i_n}^{z} \#((i_1, \ldots, i_{n-1}), z')$$
- summing up of $i_n + 1$ consecutive values of $f(x) = \#((i_1, \ldots, i_{n-1}), x)$
- illustrating example: $I = (30, 50, 10)$

$$\#((30), z) = \begin{cases} 0 & \text{if } z < 0 \\ 1 & \text{if } 0 \leq z \leq 30 \\ 0 & \text{if } z > 30 \end{cases}$$

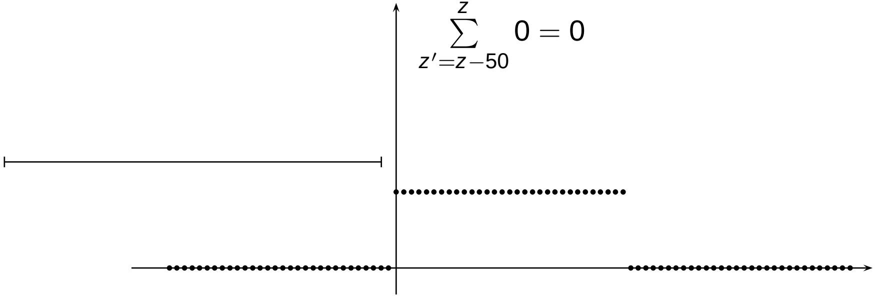


$$\#((30, 50), z)?$$



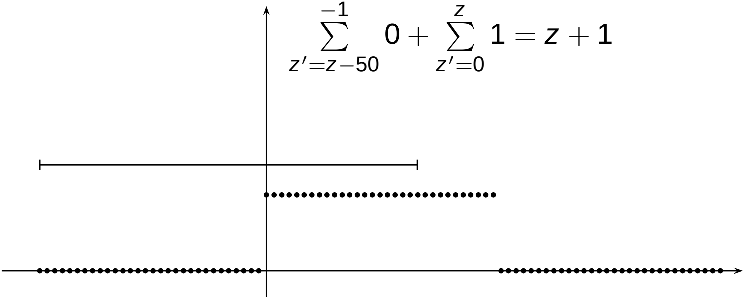
$z < 0$

$$\#((30,50),z) = \sum_{z'=z-50}^{z} \#((30),z') = \sum_{z'=z-50}^{z} 0 = 0$$



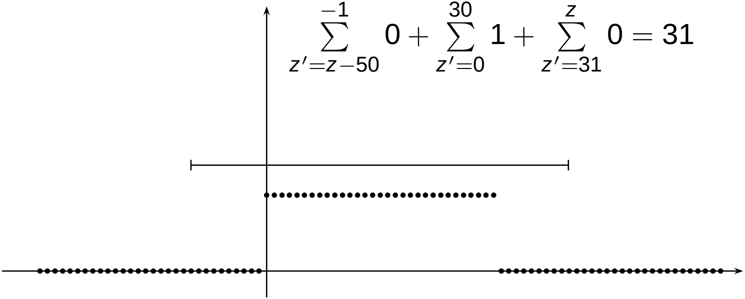
$0 \leq z \leq 30$

$$\#((30,50),z) = \sum_{z'=z-50}^{z} \#((30),z') = \sum_{z'=z-50}^{-1} 0 + \sum_{z'=0}^{z} 1 = z+1$$



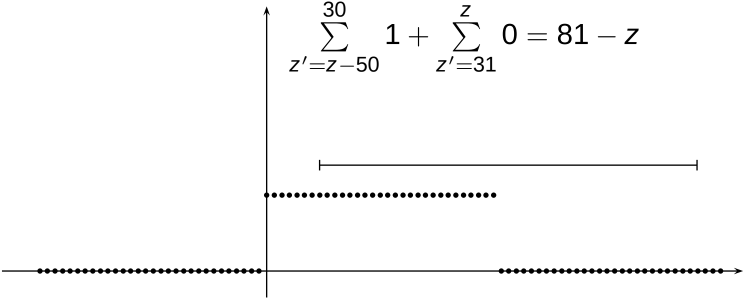
$31 \leq z \leq 49$

$$\#((30,50),z) = \sum_{z'=z-50}^{z} \#((30),z') = \sum_{z'=z-50}^{-1} 0 + \sum_{z'=0}^{30} 1 + \sum_{z'=31}^{z} 0 = 31$$

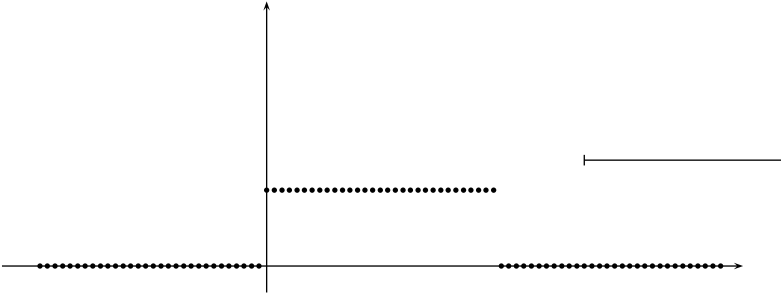


$50 \leq z \leq 80$

$$\#((30,50),z) = \sum_{z'=z-50}^{z} \#((30),z') = \sum_{z'=z-50}^{30} 1 + \sum_{z'=31}^{z} 0 = 81 - z$$

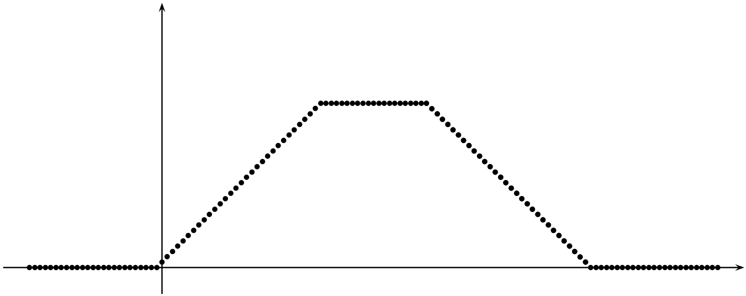


$z > 81$ $\#((30,50),z) = 0$

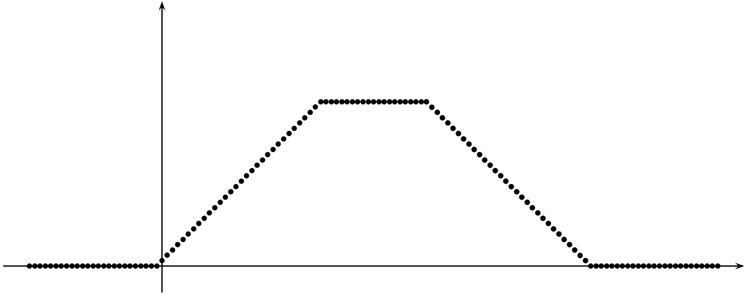


z	$\#((30,50),z)$
$-\infty < z \leq -1$	0
$0 \leq z \leq 30$	$z+1$
$31 \leq z \leq 49$	31
$50 \leq z \leq 80$	$81-z$
$81 \leq z < \infty$	0

piecewise defined polynomials of degree at most 1

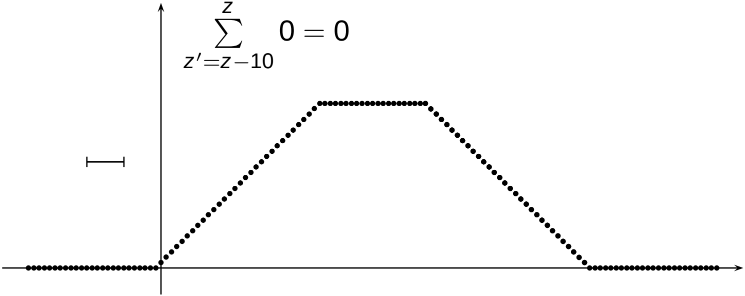


$$\#((30, 50, 10), z) = ?$$



$z < 0$

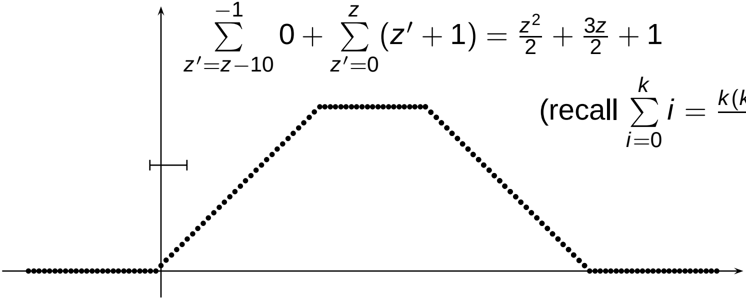
$$\#((30,50,10),z) = \sum_{z'=z-10}^{z} \#((30,50),z') = \sum_{z'=z-10}^{z} 0 = 0$$



$0 \leq z \leq 9$ $\quad \#((30,50,10),z) = \sum_{z'=z-10}^{z} \#((30,50),z') =$

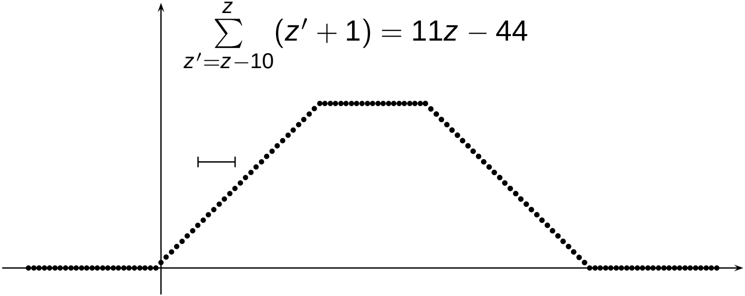
$$\sum_{z'=z-10}^{-1} 0 + \sum_{z'=0}^{z} (z'+1) = \frac{z^2}{2} + \frac{3z}{2} + 1$$

$$\text{(recall } \sum_{i=0}^{k} i = \frac{k(k+1)}{2}\text{)}$$



$$10 \leq z \leq 30 \quad \#((30, 50, 10), z) = \sum_{z'=z-10}^{z} \#((30, 50), z') =$$

$$\sum_{z'=z-10}^{z} (z' + 1) = 11z - 44$$



z	$\#((30, 50, 10), z)$
$-\infty < z \leq -1$	0
$0 \leq z \leq 9$	$\frac{z^2}{2} + \frac{3z}{2} + 1$
$10 \leq z \leq 30$	$11z - 44$
$31 \leq z \leq 40$	$-\frac{z^2}{2} + \frac{81z}{2} - 479$
$41 \leq z \leq 49$	341
$50 \leq z \leq 59$	$-\frac{z^2}{2} + \frac{99z}{2} - 884$
$60 \leq z \leq 80$	$-11z + 946$
$81 \leq z \leq 90$	$\frac{z^2}{2} - \frac{183z}{2} + 4186$
$91 \leq z < \infty$	0

piecewise defined polynomials of degree at most 2

- idea: compute $\#((i_1, \ldots, i_k), z)$ as piecewise defined polynomials from $\#((i_1, \ldots, i_{k-1}), z)$
- computation of the new validity intervals: easy
- computation of the new polynomials:
  - in the example via $\sum_{i=0}^{k} 1 = k + 1$ and $\sum_{i=0}^{k} i = \frac{i(i+1)}{2}$
  - for higher degrees via Faulhaber's formula
  - $\sum_{i=0}^{n} i^p = \sum_{k=0}^{p} \binom{p}{k} \frac{B_{p-k}}{k+1} n^{k+1}$
  - B_m denotes m-th Bernoulli number (efficiently computable)

- eventually $((i_1, \ldots, i_n), z)$ is given as piecewise defined polynomials of degree at most $n - 1$
- for concrete value of z find the associated polynomial by binary search and evaluate it
- bitter pill: number of validity intervals can be exponential in the number of upper bounds
- bounded by $2^n + 1$
- bound is tight if $i_l > \sum_{k=1}^{l-1} i_k$

Theorem

Given a fixed sequence $I = (i_1, i_2, \ldots, i_n)$ of n positive integer numbers there is a data structure which can be constructed in $O(2^n n^3)$ time and can be used to determine the value $\#(I, z)$ for every $z \in \mathbb{Z}$ in $O(n)$ time.

- in practical applications:
  - size of upper bound around 10000
  - number of upper bounds up to 10
  - in particular interested in $\#(I, \frac{\sum I}{2})$
- implemented in `C++` using `GMP`-library for big number arithmetic
- run on one core of Intel(R) Xeon(R) CPU E5540 clocked at 2.53GHz
- http://developer.berlios.de/projects/intpartition/

l, z	new algorithm	naïve algorithm
$(10^4, 10^4, 10^4, 10^4, 10^4), 1$	54 ms	4 ms
$(10^4, 10^4, 10^4, 10^4, 10^4), 300$	53 ms	302 ms
$(10^4, 10^4, 10^4, 10^4, 10^4), 2.5 \cdot 10^4$	57 ms	11 min(!)
$(10993, 10520, 10856, 10346, 10039), 1$	70 ms	4 ms
$(10993, 10520, 10856, 10346, 10039), 26377$	68 ms	13 min(!)
$(33, 29, 42, 34, 59, 76, 54, 33), 180$	345 ms	78 ms
$(10000, 10005, 10010, 10015, 10021, 10027, 10039, 10063), 40090$	458 ms	$>$ 1 h
$(10000, 10000, 10000, 10000, 10000, 10000, 10000, 10000, 10000, 10000), 50000$	74 ms	$>$ 1 h
$(10993, 10520, 10856, 10346, 10039, 10644, 10005, 10941, 10718, 10305), 52683$	4,5 s(!)	$>$ 1 h
$(12184, 12324, 14685, 11098, 13357, 13863, 10796, 10914, 10989, 11115, 10937), 66131$	17 s(!)	$>$ 1 h
$(12184, 12324, 14685, 11098, 13357, 13863, 10796, 10914, 10989, 11115, 10937, 13634), 72948$	86 s(!)	$>$ 1 h

Optimisation

- avoid superfluous computations for the single shot problem
- exploit symmetry property
- parallelisation
- use fast convolution algorithms for splitting property
  $\#((i_1, \ldots, i_n), z) = \sum_{z'=0}^{z} \#((i_1, \ldots, i_{n_1}), z') \cdot \#((i_{n_1+1}, \ldots, i_n), z - z')$

Questions?