Implicit data structures

Goldsmiths Computing

Motivation

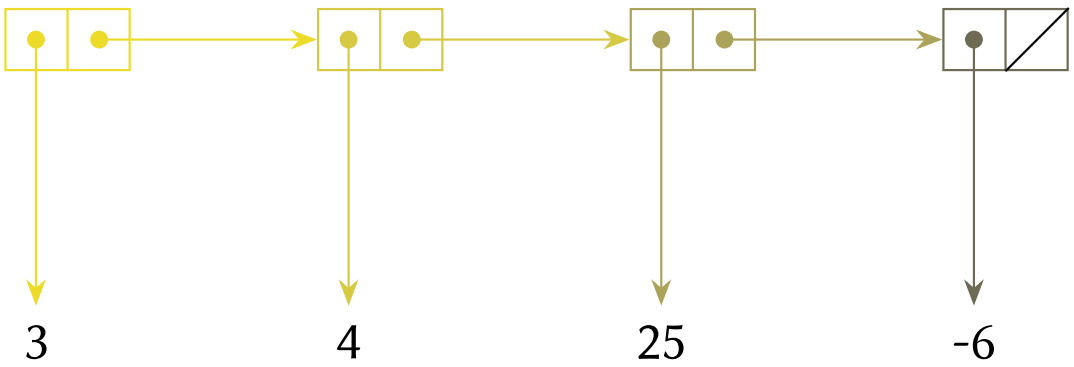
Pointers in data structures can be wasteful of space and cause inefficiencies on modern architectures. Encoding relationships (e.g. parent, left-child) between elements using storage location can help.
Pointers/references can also be hard to work with. We're not going to solve *that* problem here.

Definition

An implicit data structure is one where the space overhead for encoding the relationship between data contained in the structure is constant, regardless of the number of elements contained in the data structure.

$$S(N) \in \Theta(1)$$

Linked list (review)



Example: linked list

Space overhead is linear

$$S(N) \in \Theta(N)$$

Example: linked list

Implement as a pair of static array and counter (A,c):

first	return A[c]
rest	return (A,c+1)
set-first![o]	A[c] ← o
set-rest![l]	?

Work

1. Reading:
    - J. Ian Munro and Hendra Suwanda, *Implicit data structures for fast search and update*, Journal of Computer and System Sciences 21:2, pp.236-250 (1980)