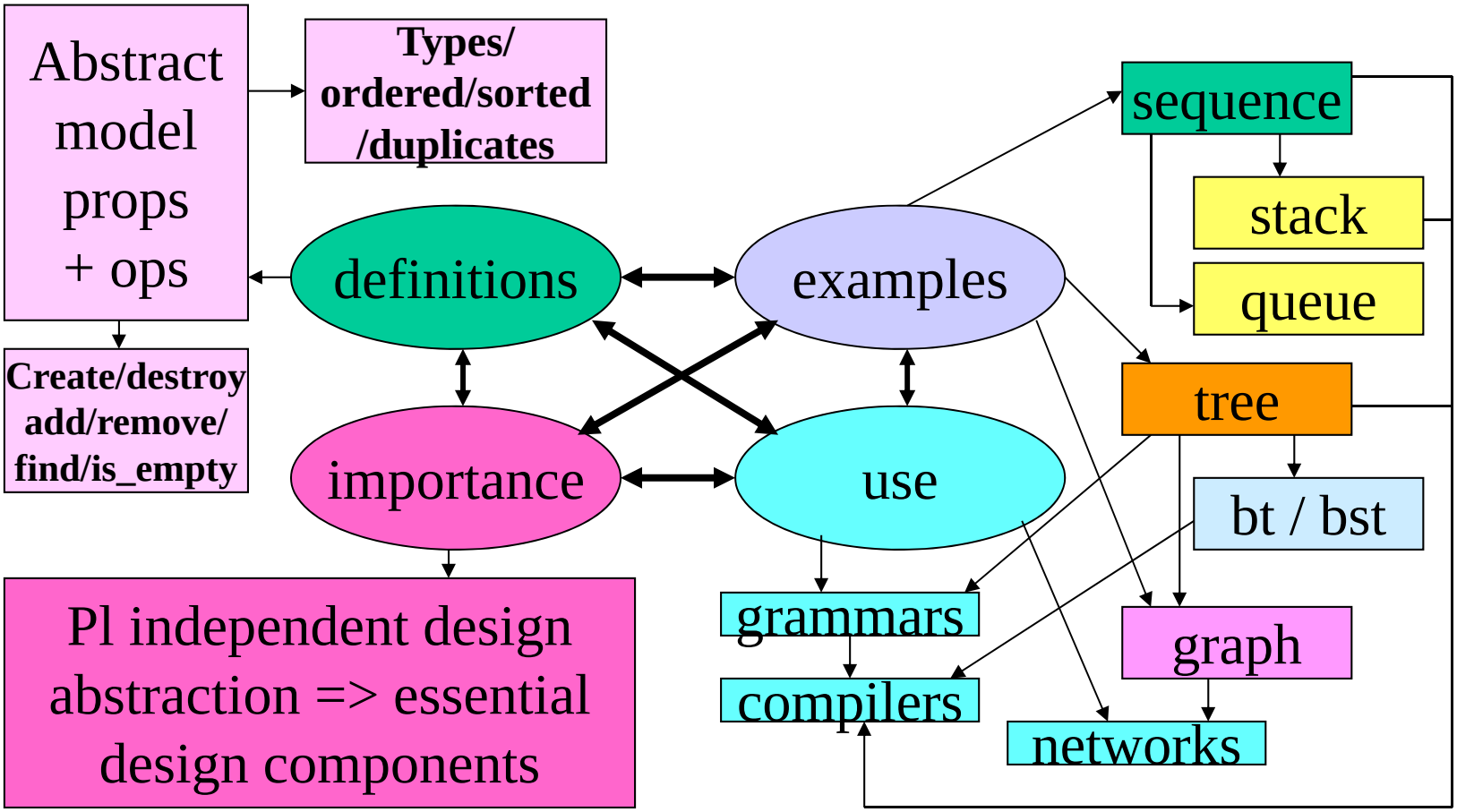
Exam question (5p)

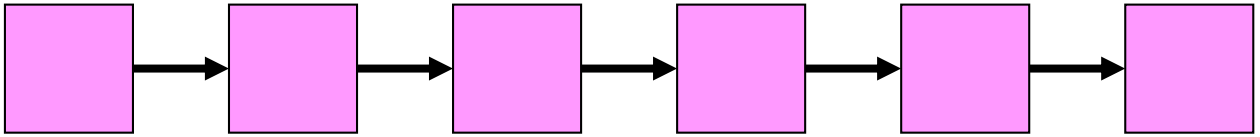
- Abstrakta datastrukturer (**ADS**) anses som ett centralt begrepp inom datavetenskap och ADS tillämpas inom flera område i datavetenskap. Förklara varför ADS är så viktiga.

- Approach (allocate possible marks to each discussion point)
  - give (a) definition(s) - e.g. use properties + operations
  - give examples of ADTs
  - give examples of use
  - why are ADTs important ?

Mind map

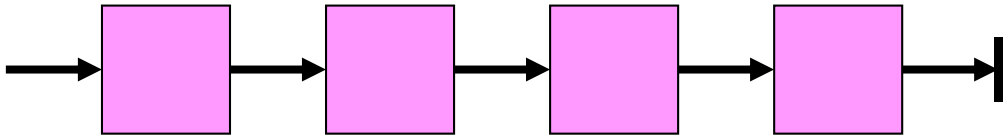

Abstract model props + ops
Types/ ordered/sorted /duplicates
Create/destroy add/remove/ find/is_empty
definitions
examples
importance
use
Pl independent design abstraction => essential design components
sequence
stack
queue
tree
bt / bst
graph
networks
grammars
compilers

Example: sequence



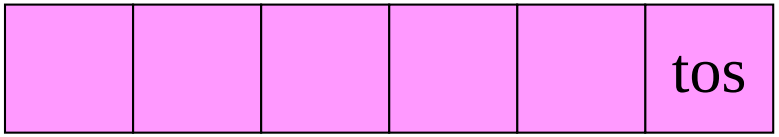
- properties: ordered / duplicates allowed / may be sorted
- operations: create / destroy / add (insert, append) / remove / find / is_empty / first / next / last / previous
- use: databases / files / text (incl. programs) / time
- importance: fundamental concept for written information / communication / thinking processes (?) / output displays

Example: linked list



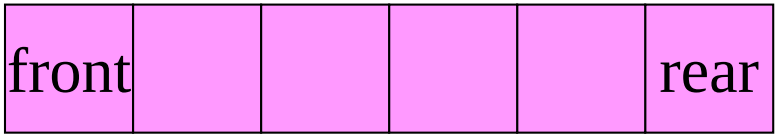
- properties: ordered / poss. sorted / duplicates allowed / recursive definition (very much related to sequences)
- operations: create / destroy / add (insert , append) / remove / find / is_empty / first / last / next / previous / recursive ops
- use: databases / files / text (incl. programs) / time
- importance: fundamental concept for written information / communication / thinking processes (?) / output displays

Example: stack



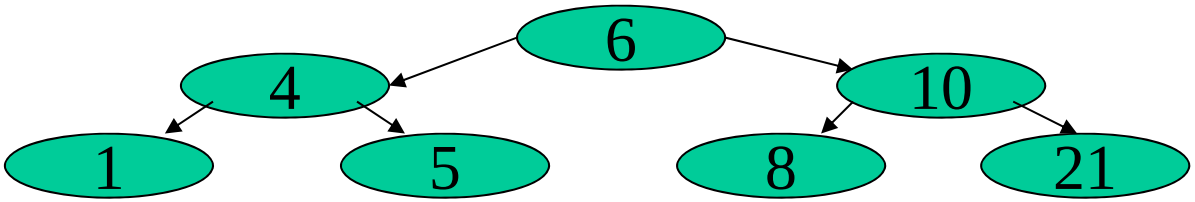
- properties: non-ordered / non sorted / duplicates allowed
- operations: create / destroy / insert (push) / remove (pop) / find (get_tos) / is_empty / first
- modified sequence ops: push = insert(S, e, first(S)), pop = remove(S, e, first(S)), get_tos = find(S, first(S))
- use: backtracking / expressions (infix => postfix) / run-time stack
- importance: basic ADT in compilers, O/Ss

Example: queue



- properties: ordered / non sorted / duplicates allowed
- operations: create / destroy / insert (enqueue) / remove (dequeue) / is_empty / first (front) / last (rear)
- modified sequence ops: enqueue = insert(S, e, first(S)), dequeue = remove(S, e, last(S)), front/rear = find(S, first(S)/last(S))
- use: in graph manipulation / breadth-first search in trees
- importance: tree / graph operations

Example: binary search tree

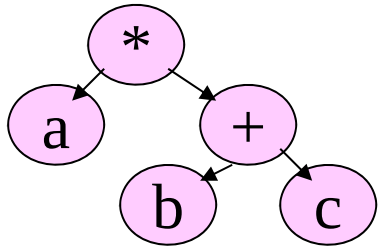


- properties: ordered / sorted / duplicates allowed / value(L-bst) <= value (node) <= value(R-bst) / insert / remove / find are O(log n) / recursive definition
- operations: create / destroy / add / remove / find / is_empty / root

  pre-, in-, post-order, breadth-first traversals (navigation) / recursive ops
- use: (trees) in compilers (expression trees) / hierarchical systems (e.g. file) / database indexing (B-trees) / program flow analysis
- importance: fast easily implemented ADT

Example: expression handling

- Infix => postfix mapping (using a stack)
  - e.g. a * (b + c) => a b c + *
- postfix => expression tree mapping (using a stack)
  - e.g. a b c + * =>



- code generation (using the tree)

```
load b, R1          t1  b  c  +
load c, R2          t2  t1 a  *
add
load a, R2          (3 address code)
mul
```

Why are ADTs so important?

- allow a large number of systems to be abstracted
- allow the implementation of most systems using a small number of well understood ADTs

  (set, sequence (list), tree, graph)
- allow the use of a more formal system (mathematics)
- provide simple conceptual models for design and analysis of systems
- existing (efficient) algorithms may be (re-)used

All that for (just!) 5p !!!

- NO - this was a fairly complete example (there are still other possible (correct) answers!)
- It is my experience (25 years at Kau) that most students tend to write too little in exam questions - it helps to give details, examples, use pictures and add explanative text
- “Proof by vigorous hand-waving” and too few details i.e. vague answers, will not get marks
- “I knew it but couldn’t express my answer in the exam” - you have a problem! Learn to express ideas in written form - this is one of the most important aspects of studying!