Basic data structures

Svarny Petr

Katedra logiky FF UK

24. března 2022

Overview

Teorie datových struktur

Array, pole

Stack, zásobník

Queue, fronta

Table, tabulka

List, seznam

Set, množina

Graph, graf

Tree, strom

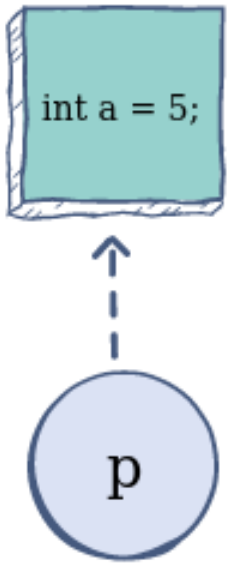
Datové typy (DT)

- **Základní** např. int, str ...
- **Ukazele a reference**, viz. =Utah=
- **Strukturované** např. pole, dict, ... → datové struktury

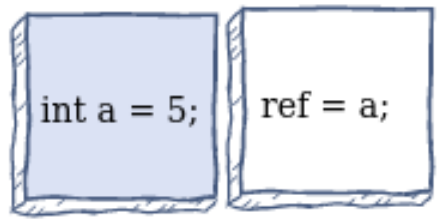
Pointers and references

https://xkcd.com/138/

Pointers and references



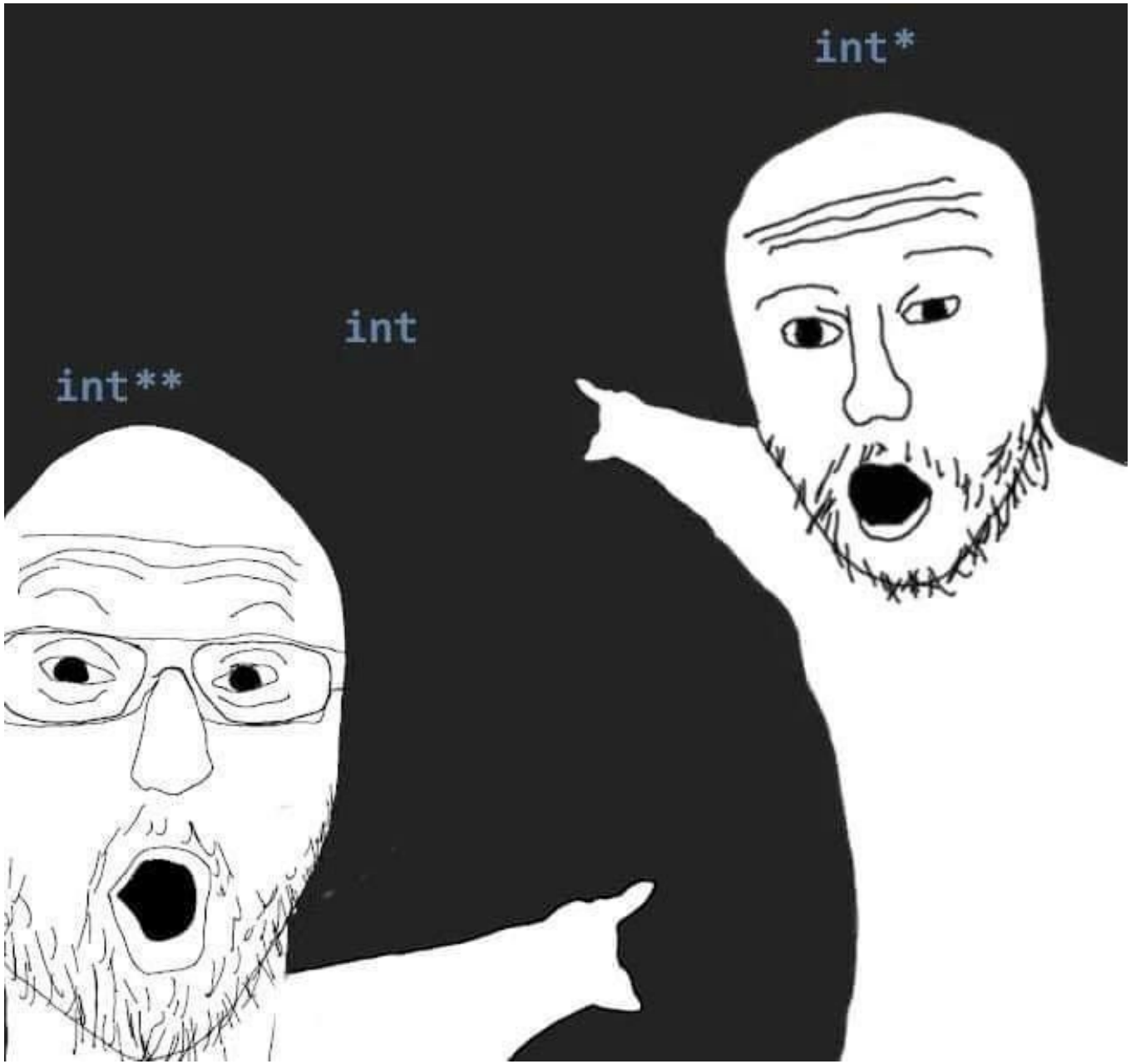
The pointer variable **p** stores the address of the variable **a**; it "points" to the memory location of **a**.



Creating a reference to **a** just makes an alias for it; it does not "point" to **a** by storing its address in a separate memory location

=Educative=

Pointers and references



Základní operace na DS

- **Traversing** - procházení
- **Searching** - prohledávání
- **Insertion** - vkládání prvku
- **Deletion** - odstranění prvku
- **Sorting** - třídění, řazení
- **Merging** - slučování
- a další (např. size, min, max)

Abstraktní DT

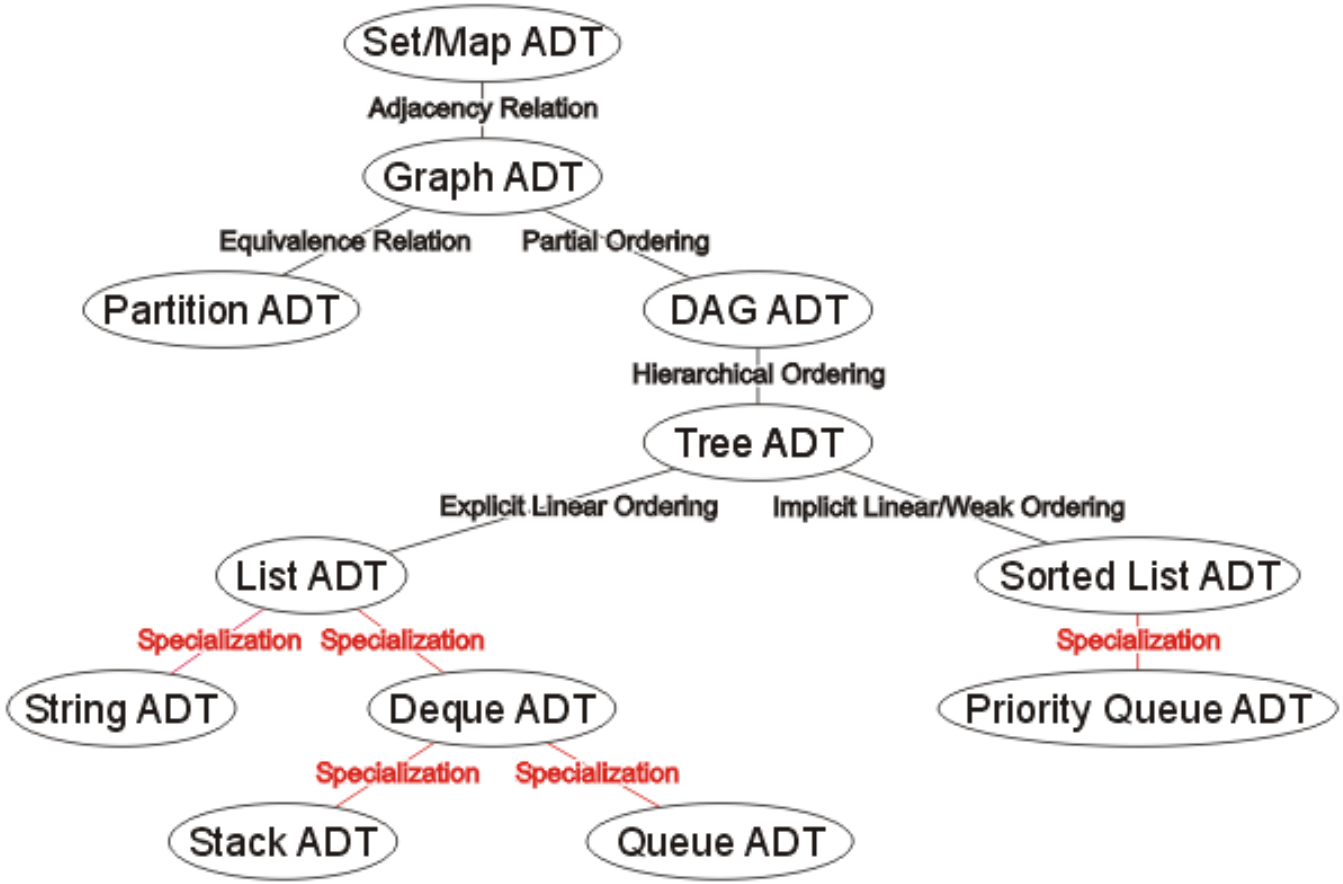
- Množina druhů dat a operací na nich
- Abstraktní, nezávislý na implementaci
- Model složitějších DT
- Definice formálně (axiomaticky) nebo programátorsky jako rozhraní (interface)

Základní ADT

- Sekvenční (lineární, má následníka)
  - Pole (Array)
  - Zásobník (Stack)
  - Fronta (Queue)
- Asociativní (nelineární)
  - Tabulka (Table, Map)
  - Množina (Set)
  - Strom (Tree)

Viz např. =UniWaterloo=

Základní ADT



=UniWaterloo=

Array - pole

- Homogenní = prvky stejného typu.
- Všechny prvky současně v paměti - random access přes indexy.
- Statický = fixní počet prvků.
- Lineární = indexy uspořádány.
- Vše je de facto 1 dimenzionální.

XD pole

Jak mohu mít více dimenzionální pole? (např. matici)

Mapovací funkce a XD pole

- $map(i, j) = a(i_{min}, j_{min}) + (i - i_{min})nj + (j - j_{min})$
- Možno udělat také pro 3D nebo řazení po sloupcích.

Python list

Je Python list pole?
Přitom:

- Může obsahovat různé typy.
- Může měnit svou velikost.

Python list

=PythonDocs=

- Alokováno pole vždy větší
  $(1.125 * \text{velikost} + 6 (+3, \text{pro pole } < 9 \text{ prvků}))$.
- Pole je pole ukazatelů.

Stack, zásobník

- A.K.A. Last In First Out
- Např. návrat z procedury, DFS průchod stromem.
- Přístup pouze k vrcholu (top).
- Dynamický = proměnlivý počet prvků.
- Lineární, Homogenní.
- Vložený prvek odkazuje na dřívější prvek.

Queue, fronta

- A.K.A. First In First Out
- Např. fronty úloh, BFS průchod stromem.
- Přístup pouze prvku na čele (head).
- Vkládání pouze na konec (tail).
- Homogenní, lineární, dynamický.
- Vložený prvek odkazuje na poslední prvek.

Table, tabulka

- A.k.a. Look-up table, vyhledávací tabulka.
- Homogenní, i dynamická, nelineární.
- Klíče (jedinečné) a asociované (případné) hodnoty.
- Vyhledávání může být implementováno:
  - Asociativní, t.j. hledané = klíč, ($\Omega(logn)$, ale $O(n)$)
  - Adresní - indexace klíčem (přímý přístup) nebo rozptylováním (hashing = výpočet adresy z klíče), $\tilde{\Theta}(1)$
- Tedy implementačně může mít mnoho jmen: map, hash table, hashmaps, associative array.

Hashing

- Kompromis, přímý přístup má prostorově $O(n)$.
- Hash má konstantní čas pro vyhledávání a vložení.
- Užívá rozptylovací (hashovací) fce (podrobněji možná příště, viz =Jenkins=).
- Ale ...
  - Konstantní čas odpovídá délce klíče.
  - Nevhodné na výběr x-tého prvku a řazení.

Python dict

Je Python dict tabulka?

Python dict

=PythonDocs=

- Dokonce vyloženě hash table.

List, seznam

- Homogenní, dynamický, lineární.
- Nemá indexovaný přístup.
- Vkládání a mazání v místě ukazovátka.

Python list

Je Python list list?

Python list

- Ne, viz dříve. Je to pole referencí.

Set, množiny

- Map, kde vše má jen klíč.
- Homogenní, dynamický nebo statický, nelineární.
- Existuje také multimnožina (multiset, bag).

Python set

Je Python set set?

Python set

- Ano, má i podobnou syntax jako dict.

Graph, graf

- Složen z uzlů a (ne)orientovaných hran.
- Realizován vícero způsoby hlavní je myšlenka sousedství (adjacency).

Tree, strom

- Speciální druh grafu.
- Každý uzel kromě kořene má jednoho rodiče/předchůdce (kořenový strom).