

ZALG 1. cvičení

Soubory z cvičení

- <https://github.com/martinnovaak/ZALGcv>

Zápočet

- Vypracování zápočtové úlohy a protokoly na domluvené téma

Asymptotická složitost

- O-značení
- $\forall g(n): O(g(n)) := \{f(n) \mid (\exists c, n_0 \in \mathbb{R})(n \geq n_0)(0 \leq f(n) \leq c \cdot g(n))\}$

Prostorová složitost

- T-značení

Objektově orientované programování

- Datový typ struktura
 - Přímě obsahuje (definuje) datové složky
 - (Bokem) definuje funkce, které s danými datovými složkami pracují
- Objektový datový typ (třída)
 - Přímě obsahuje datové složky
 - Přímě obsahuje (definuje) i operace nad objekty dané třídy

Terminologie

- Třída – klíčové slovo class
 - Objektový datový typ
- Instance (objekt) – proměnná objektového datového typu
- Složky třídy:
 - Atributy – datové složky
 - Metody – funkce
 - Konstruktor – postará se o vytvoření a inicializaci nové instance (např. u seznamu se nemusí volat zvlášť funkce vytvořSeznam())
 - Destruktor – postará se o zrušení instance
 - Instanční – metoda pro práci s jednotlivými instancemi
 - Třídní – metody pro práci se třídou jako celkem

Konstruktor

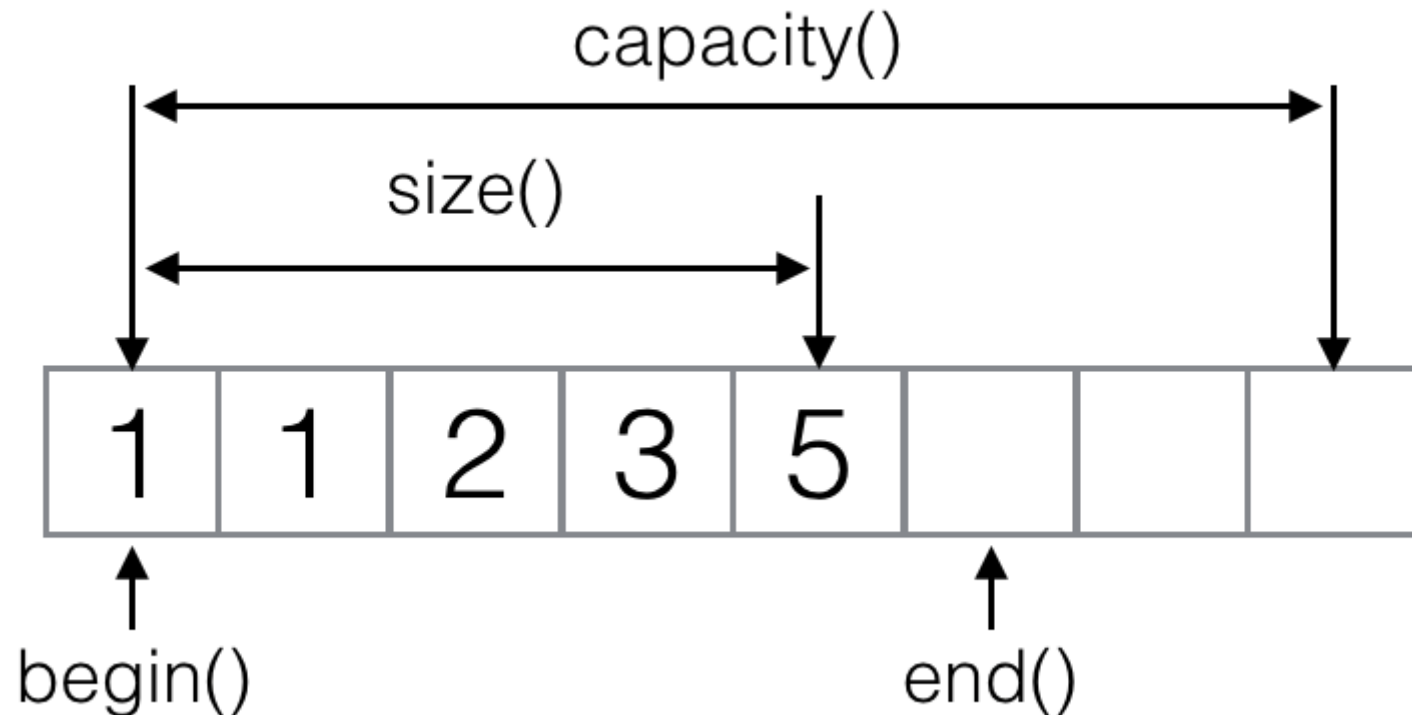
- Metoda sloužící k (vytvoření) inicializaci instance
- Nevolá se přímo, je volán automaticky
 - Při vytvoření instance
- Jmenuje se stejně jako třída
- Neobsahuje návratovou hodnotu
- Může obsahovat libovolné parametry s výjimkou své třídy (Ale může obsahovat referenci na svou třídu – u kopírovacích konstruktorů)
- Pokud nedeklarujeme konstruktor, tak překladač vytvoří implicitní konstruktor bez parametrů

Destruktor

- Speciální metoda sloužící ke zrušení instance
 - Uvolní dynamicky alokované paměti
 - Uzavře soubory, s nimiž třída pracuje atd...
- Je volán automaticky při zániku instance
- Jmenuje se stejně jako třída, před identifikátorem obsahuje ~
- Neobsahuje návratovou hodnotu a nesmí mít parametry
- Každá třída smí mít maximálně jeden destruktork

Chytré pole - Vector

- Chytré dynamické pole. Prvky jsou v paměti uloženy za sebou.
- Vector se podle potřeby umí sám rozšiřovat.

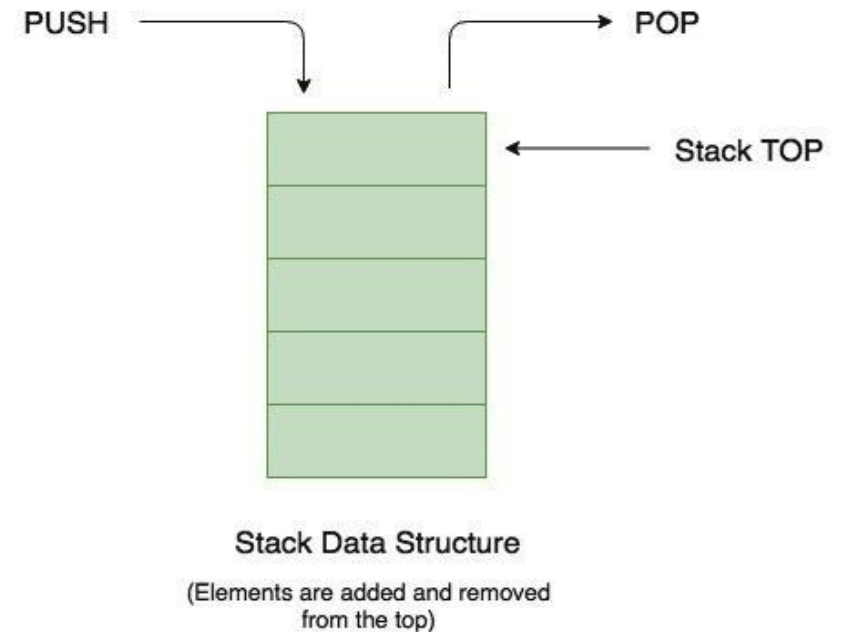


Spojový seznam

- Datová struktura, která představuje posloupnost datových položek neboli prvků
- Uspořádaná datová struktura, ale v paměti nemusí prvky ležet za sebou
- Metody:
 - Vytvoření prázdného seznamu
 - Přidání prvku na konec seznamu
 - Vyhledání údaje v seznamu
 - Odstranění prvního prvku
 - Odstranění posledního prvku

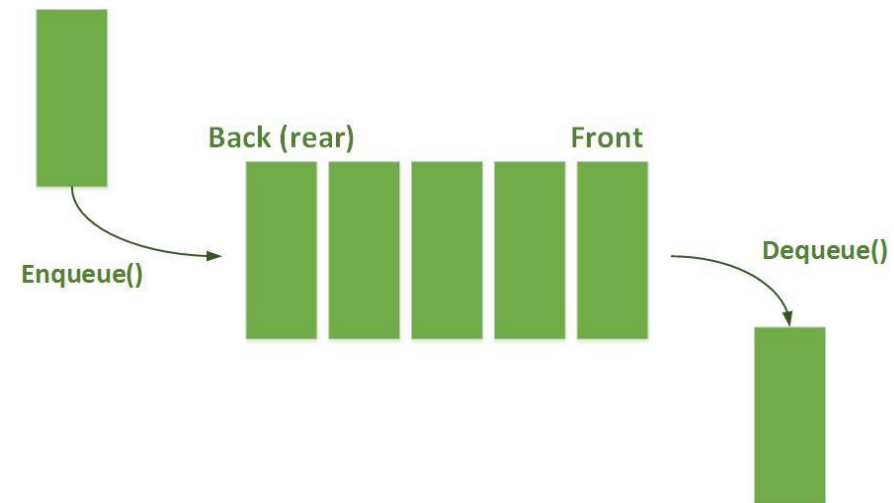
Zásobník – Stack

- Princip LIFO – last in first out
- Jedná se o tzv. adaptér
- Kontejner se dvěma základními operacemi: vyjmutí a vložení. Vyjme se ten prvek, který byl vložen poslední.
- Můžeme ho implementovat pomocí pole nebo seznamu



Fronta - Queue

- Princip FIFO – first in, first out
- Adaptér
- Kontejner se dvěma základními operacemi: vložení hodnoty a vyjmutí
- Položky vyjímáme z fronty, ve kterém jsme je do ní vložili
- Prvek z fronty můžeme vyjmout jen tehdy, je-li prvek na řadě, ale jeho hodnotu můžeme přečíst kdykoli

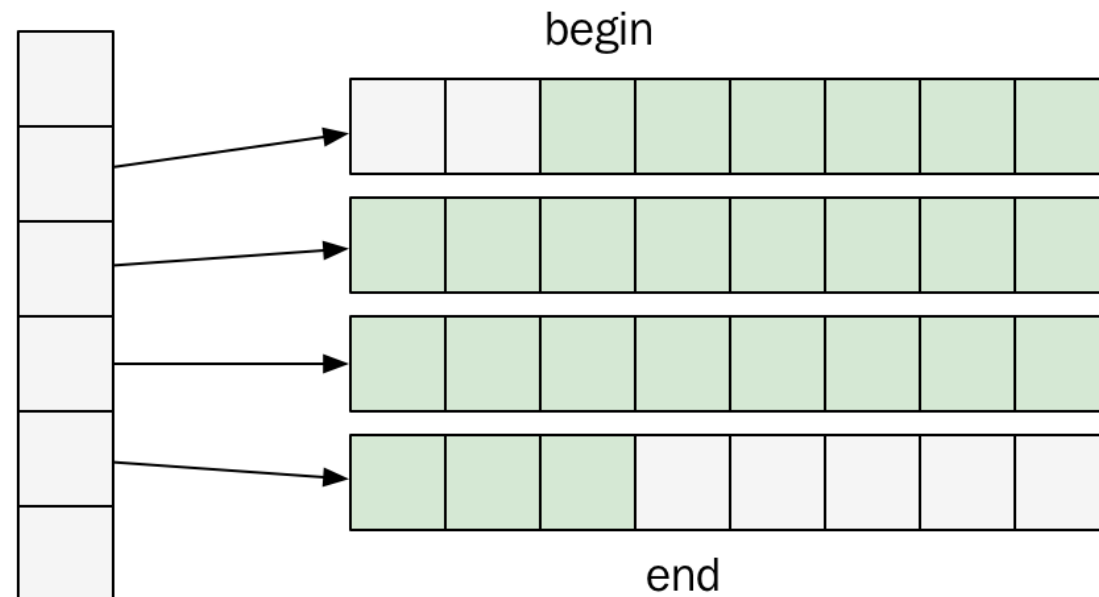


Prioritní fronta – Priority Queue

- Fronta s předbíháním
- Od fronty se liší tím, že v ní jsou prvky seřazeny podle priorit. Vkládáme je v pořadí, ve kterém přišly, ale vybíráme je na základě jejich priority – prvně prvek s nejvyšší prioritou.
- V STL knihovně reprezentovaná pomocí binárního stromu

Obousměrná fronta - Deque

- Indexovaný sekvenční kontejner, který kromě rychlého vkládání na konec umožňuje rychlé vkládání na začátek.



Stl knihovna

- Vektor – `std::vector<T>`
- Spojový seznam – `std::list<T>`
- Fronta – `std::queue<T>`
- Zásobník – `std::stack<T>`
- Prioritní fronta – `std::priority_queue<T, "Komparátor">`

Diskuze k probraným datovým strukturám

- Implementace
- Asymptotická složitost příkazů
 - Náhodný přístup – Random access
 - Vkládání prvku na začátek – push_front
 - Vkládání prvku na konec – push_back
 - Odstranění prvku z počátku – pop_front
 - Odstranění prvku z konce – pop_back

Příklad 1: porovnání STL knihoven

Příklad 2: Vlastní zásobník a fronta

Dobrovolný domácí úkol

- Prioritní fronta přes spojový seznam:
 - Metoda vlož bude upravena tak, aby vkládala do seznamu na správné místo