

Domáce zadanie 3 – analýza výkonu ADT prioritný front

Cieľom domáceho zadania je porovnanie dvoch implementácií ADT prioritný front. V rámci testovania je nutné porovnať výkonnosť štruktúr v konkrétnych scenároch a odhadnúť časovú zložitosť konkrétnych operácií v závislosti na počte prvkov v zozname.

Úloha 1 – implementácia

Implementujte ADT prioritný front, ktorý musí podporovať operácie definované v súbore **priority_queue.h**. Implementáciu realizujte ako nasledujúce AUŠ:

- *ľavostranná halda* (**heap.h**);
- *dvojzoznam* (**priority_queue_two_lists.h**).

V triedach definovaných v súboroch **heap.h** a **priority_queue_two_lists.h** je nutné implementovať všetky verejné metódy a podľa potreby si môžete pridať ľubovoľné atribúty a ľubovoľné súkromné metódy. **Môžete využiť vlastnú implementáciu z cvičení.** Pre implementáciu dvojzoznamu môžete s veľkou výhodou využiť existujúce implementácie utriedeného a neutriedeného sekvenčného prioritného frontu, ktoré môžu reprezentovať jednotlivé zoznamy v dvojzozname (podrobný návod nájdete v prezentácii k cvičeniu 6 od snímky 30).

Správnu implementáciu jednotlivých operácií je nutné otestovať prostredníctvom Vami implementovaných jednotkových testov (unit testov). Tieto implementujte v súboroch **priority_queue_test.h** a **priority_queue_test.cpp**.

Súbory **priority_queue.h**, **heap.h**, **priority_queue_two_lists.h**, **priority_queue_test.h** a **priority_queue_test.cpp** nájdete v projekte využívanom na cvičeniach.

Úloha 2 – overenie výkonu v scenári

Realizácie ADT prioritný front, ktoré ste implementovali v úlohe 1, otestujte v scenároch definovaných v Tab. 1. V každom scenári vykonajte spolu 100 000 operácií. Jednotlivé operácie sú v jednotlivých scenároch volané náhodne tak, aby na konci súhlasil podiel jednotlivých operácií (neplatí, že najskôr sa zavolajú operácie na vkladanie prvkov, potom operácie na mazanie prvkov a nakoniec operácie na sprístupnenie prvkov). Parametre do operácií sú taktiež náhodné; priorita je náhodné číslo z intervalu $<0; 100\,000>$.

Tab. 1 Testovacie scenáre pre ADT prioritný front

Scenár	Podiel operácií		
	push	pop	peek
A	35	35	30
B	50	30	20
C	70	25	5

V rámci analýzy výkonu v scenári je nutné merať len dĺžku trvania vybranej operácie. To znamená, že do merania **sa nesmie započítavať čas potrebný pre generovanie pomocných údajov**.

Úloha 3 – analýza časových zložitostí

V rámci analýzy časových zložitostí je nutné otestovať rýchlosť operácií **push**, **pop** a **peek** v závislosti od počtu prvkov a implementácie prioritného frontu a na základe nameraných a spracovaných údajov **odhadnúť hornú asymptotickú zložitost' jednotlivých operácií**.

V rámci analýzy časových zložitostí vybraných operácií je nutné merať len dĺžku trvania vybranej operácie. To znamená, že do merania **sa nesmie započítavať čas potrebný pre generovanie pomocných údajov**.

Úloha 4 – bonus

Cieľom bonusovej úlohy je otestovať vplyv dĺžky kratšieho zoznamu na výkon prioritného frontu implementovaného ako dvojzoznam (využite implementáciu z úlohy 1). Pri testovaní je nutné porovnať nasledujúce stratégie pre definovanie dĺžky kratšieho zoznamu (či už pri vzniku štruktúry alebo pri jej reštrukturalizácii):

- konštantná dĺžka definovaná ako 1/1000 počtu všetkých vložení do prioritného frontu;
- variabilná dĺžka definovaná ako \sqrt{n} , kde n je počet prvkov v prioritnom fronte;
- variabilná dĺžka definovaná ako $n/2$, kde n je počet prvkov v prioritnom fronte.

Uvedené stratégie porovnajte s využitím scenárov definovaných v úlohe 2. Pre umožnenie menenia dĺžky kratšieho zoznamu na základe rôznej stratégie môžete využiť podobné prístupy, ako boli uvedené v bonusovej úlohe v domácom zadaní 2.

Výstupy domáceho zadania

Domáce zadanie bude mať nasledujúce výstupy:

1. súbory (*odovzdávate všetky súbory bez ohľadu na to, ktoré časti domáceho zadania riešite*):
 - o **heap.h** – implementácia ADT prioritný front – ľavostranná halda (úloha 1);
 - o **priority_queue_two_lists.h** – implementácia ADT prioritný front _dvojzoznam (úloha 1) – ak využívate aj ďalšie zdrojové súbory (napr. `priority_queue_limited_sorted_array_list.h` ak implementujete dvojzoznam podľa návodu), odovzdajte aj tie;
 - o **priority_queue_test.h**, **priority_queue_test.cpp** – implementácia jednotkových testov ADT prioritný front (úloha 1) vrátane výkonnostných testov definovaných v úlohách 2, 3 a 4;
2. CSV súbory s údajmi zaznamenanými počas testov (*odovzdávate len v prípade riešenia úloh 2, 3 a 4*);
3. dokumentácia (*odovzdávate len v prípade riešenia úloh 2, 3 a 4*), ktorá obsahuje:
 - o popis realizácie priebehu jednotlivých testov – v prípade úlohy 2 je nutné vysvetliť, ako budete realizovať jednotlivé scenáre, t. j. ako sa budete v danom scenári rozhodovať, ktorú operáciu vykonáte;
 - o popis formátu údajov a CSV súborov v testoch – je nutné popísať údaje, zaznamenávané počas testov, a formát CSV súboru (teda, čo sa bude v CSV súbore nachádzať a kde), do ktorého budete tieto údaje ukladať;
 - o metodiku spracovania a vyhodnotenia výsledkov testov (CSV súborov) – je nutné popísať, ako budete analyzovať výsledné CSV súbory;
 - o prezentáciu výsledkov a záverov vyplývajúcich z testovania.

Požadované výstupy zabaľte do archívu. Ak sa stane (podľa počtu úloh a charakteru zbieraných dát), že výsledný archív po zabalení bude mať kvôli CSV súborom viac ako 15 MB, tak ho nebude možné odovzdať cez systém moodle. V takomto prípade do archívu nepridajte CSV súbory, avšak zachovajte si ich. Ak počas čítania dokumentácie narazíme na nezrovnalosť, tak vás kontaktujeme a súbory nám dodatočne pošlete napr. cez MS Teams. Samozrejme, ak robíte iba bod 1, žiadne CSV súbory nevytvárate a odovzdávate iba zdrojové súbory.

V kóde používajte namiesto číselných konštánt symbolické. Údajové štruktúry a testy musia byť naprogramované tak, aby plne dodržiavali rozhrania špecifikované v súboroch **heap.h**, **priority_queue_two_lists.h** a **priority_queue_test.h**, čím ich bude možné bez problémov zakomponovať do projektu AUS používanom na cvičeniach. Údajové štruktúry a testy musia byť Vami naprogramované a **efektívne** implementované z pohľadu výpočtovej a pamäťovej zložitosti. Po ukončení behu projektu AUS s Vašimi súbormi musí byť pamäť **preukázateľne čistá** (nevznikli „memory leak-y“).

Testy vyprodukujú CSV súbor so zaznamenanými údajmi. Vyprodukované CSV súbory je potrebné spracovať (akokoľvek – ručne, pomocou tabuľkového editora, pomocou ďalšej alebo tej istej aplikácie). **Metodické popísanie spôsobu spracovania údajov je súčasťou dokumentácie.** Pomocou popísaného postupu musí byť možné nové údaje spracovať rovnako ako pôvodné. Výsledky spracovania (grafy, priemery, odhady, atď.) je potrebné

zdokumentovať a interpretovať (vyvodiť závery). Identifikácia relevantných veličín je súčasťou domáceho zadania.

Bodovanie domáceho zadania

Úloha	Ľavostranná halda	Dvojzoznam	Podmienka
1 – implementácia	2	2	-
2 – overenie výkonu v scenári	1	1	správna implementácia štruktúry
3 – analýza časových zložítostí	2	2	správna implementácia štruktúry
4 – bonus	1		musí byť všetko vyššie pre obe štruktúry

Aby bolo možné získať uvedený počet bodov za jednotlivé funkcionality, je nutné s každou funkcionalitou odovzdať aj príslušnú časť dokumentácie (okrem bodu 1).