

Projektowanie algorytmów i metody sztucznej inteligencji - Raport 1

Damian Ryś

26 kwietnia 2021

Spis treści

1	Wstęp	1
2	Zadanie	2
3	Kolejka Priorytetowa	2
3.1	Czym jest kolejka priorytetowa?	2
3.2	Zalety i wady	2
3.3	Implementacja	3
3.4	Złożoności obliczeniowe	3
3.5	Testy	3
4	Wnioski	3
5	Bibliografia	3

1 Wstęp

Grupa lab: E12-93c zmien

Termin zajęć: PN 18:45-20:35

Numer indeksu: 252936

Prowadzący: Dr inż. Piotr Ciskowski

2 Zadanie

Postawionym przed nami problemem jest wysłanie przez użytkownika A do użytkownika B wiadomości przez Internet. wiadomość ta powinna zostać wysłana w serii pakietów na komputer użytkownika B. Problem polega w tym, że pakiety mogą przychodzić z różnych przyczyn w losowej kolejności do użytkownika B. Nasz program powinien zatem:

1. Podzielić naszą wiadomość na szereg pakietów
2. Przelosować pakiety w celu zasymulowania sytuacji
3. Wczytać do struktury danych nasze pakiety
4. Przesyłać pakiety w formacie [klucz, wartość]
 - (a) klucz powinien mieć unikalną wartość
 - (b) wartość powinna być dowolnego typu (użytkownik powinien mieć możliwość przesłania dowolnej wiadomości)
5. Złączyć wiadomość w jedną, prawidłową całość i wyświetlić użytkownikowi

3 Kolejka Priorytetowa

Wobec postawionego problemu strukturą ADT, którą wybrałem jest kolejka priorytetowa bazowana na drzewie binarnym.

3.1 Czym jest kolejka priorytetowa?

Kolejka priorytetowa (ang. priority queue) jest kolejką, w której elementy są ułożone nie w kolejności wprowadzania, lecz w kolejności priorytetu. Każdy element kolejki posiada dodatkowe pole priority, w którym przechowuje swój priorytet – czyli ważność. Gwarantuje to pobieranie z kolejki jako pierwszych elementów o najwyższym priorytecie. Elementy o priorytetach niższych zostaną pobrane dopiero wtedy, gdy zostaną usunięte wszystkie elementy o priorytetach wyższych.

3.2 Zalety i wady

Strukturę tą wybrałem ze względu na:

- Łatwość implementacji
- Złożoność czasową lepszą niż $O(n)$ dla uzyskania min oraz max
- Dynamiczną alokację danych

3.3 Implementacja

//kod klasy reszta kodu dostepna na githubie

3.4 Złożoności obliczeniowe

Dla naszego programu zgodnie z założeniami projektowymi zostały obliczone złożoności czasowe poszczególnych funkcji naszej struktury: //przykład

Tak przedstawia się reszta złożoności obliczeniowych naszej struktury: //tabelka

Nasze wyniki pokrywają się z danymi dostępnymi w internecie // jakiś ref
// ,zatem uznaje , że kolejka została zaimplementowana prawidłowo.

3.5 Testy

Za bazę naszych testów posłużyły nam "polskie copy pasty". Teksty zostały podzielone,następnie wymieszane, załadowane do struktury, by finalnie zostały posortowane i połączone w jedną całość w osobnym pliku wynikowym. Na poniższym przykładzie został zaprezentowany cykl życia naszego pliku.

Program finalnie przeszedł testy na bazie większej ilości plików i był w stanie prawidłowo odtworzyć plik, zatem uznaje, że program działa prawidłowo.

4 Wnioski

5 Bibliografia

<https://bradfieldcs.com/algos/trees/priority-queues-with-binary-heaps/> <https://www.hackerearth.com/practice/data-structures/priority-queues/>