STRUKTURY BAZ DANYCH – PROJEKT NR 2

Damian Strojek, s184407

1 Wprowadzenie

Celem projektu było zaimplementowanie wybranej indeksowej organizacji pliku. W moim przypadku jest to organizacja pliku indeksowo-sekwencyjna. Metoda ta wykorzystuje dwa pliki. Pierwszy, nazywany plikiem danych, zawiera w każdym wierszu klucz, rekord oraz wskaźnik na kolejny element. Plik ten składa się z dwóch części. Pierwsza część, podstawowa (*primary area*), podzielona jest na strony o stałym rozmiarze. Druga część pliku danych nazwana jest nadmiarową (*overflow area*) i zawiera dane, które nie zmieściły się na stronie w przestrzeni podstawowej. Dane na każdej stronie w obszarze podstawowym oraz w przestrzeni nadmiarowej są posortowane rosnąco według wartości klucza. Jako klucz obrałem cześć całkowitą z wartości według której sortowałem rekordy w zadaniu pierwszym, czyli mocy (iloczyn napięcia i natężenia). Wskaźnik w przestrzeni podstawowej wskazuje na kolejny, pod względem wartości klucza, element, który nie zmieścił się w tej przestrzeni i został wstawiony do obszaru przepełnienia. Wskaźnik w przestrzeni nadmiarowej wskazuje na kolejny element w tejże przestrzeni, który powinien zostać dodany w tym samym miejscu, jednak z powodu braku miejsca na stronie również znalazł się w przestrzeni nadmiarowej. Drugi plik – indeks – zawiera indeks rzadki stworzony z kluczy z przestrzeni podstawowej pliku danych. Oznacza to, iż każdy wiersz w indeksie zawiera klucz pierwszego elementu z każdej podstawowej strony pliku danych oraz numer strony, na której znajduje się ten klucz.

2 Realizowane operacje

Program realizuje wymienione operacje:

- Wstawianie rekordów (*add vol amp*) – za pomocą indeksu poszukiwany jest numer strony, do której powinien zostać dodany rekord, następnie sprawdzamy czy na stronie są wolne miejsca. Jeśli tak to dodajemy do niej rekord, jeśli nie to dodajemy rekord do przestrzeni nadmiarowej. Po wstawieniu strona lub przestrzeń nadmiarowa jest sortowana. Podczas tej operacji należy pamiętać o sprawdzeniu czy w pliku nie znajduje się już rekord o podanym kluczu oraz o odpowiednim ustawieniu wskaźników.
- Usuwanie rekordu (*del key*) – za pomocą indeksu poszukiwany jest numer strony, na której powinien znajdować się rekord z wczytanym kluczem, następnie sprawdzamy czy rekord znajduje się w przestrzeni podstawowej, jeśli nie to należy znaleźć rekord o mniejszym kluczu od niego i za pomocą wskaźników odnaleźć szukany element w przestrzeni nadmiarowej. Jeżeli rekord z podanym kluczem znajduje się w pliku, należy go zastąpić wolnym rekordem, posortować przestrzeń, w której się znajdował oraz odpowiednio pozmieniać wskaźniki innych rekordów.
- Aktualizacja rekordu (*upd key vol amp*) – wykorzystujemy tutaj dwie wcześniej zdefiniowane operacje. Najpierw próbujemy usunąć rekord z podanym kluczem, jeśli ta operacja się powiedzie przeprowadzamy operacje wstawiania nowego rekordu z wcześniej wczytanymi danymi.

- Odczyt rekordu (*read key*) - za pomocą indeksu poszukiwany jest numer strony, na której powinien znajdować się rekord z wczytanym kluczem, następnie szukamy rekord na tej stronie. Jeżeli rekord znajduje się na stronie to wyświetlamy dane, jeżeli nie to za pomocą wskaźników poprzedzających go elementów szukamy tego rekordu.
- Reorganizacja pliku (*reorg*) - zarówno przestrzeń podstawowa, jak i nadmiarowa, ma określoną wielkość. Gdy wielkość ta zostanie wykorzystana (lub na żądanie użytkownika) zostaje przeprowadzona reorganizacja pliku. Reorganizacja polega na stworzeniu nowych stron w przestrzeni podstawowej i umieszczeniu w nich danych zarówno z przestrzeni podstawowej, jak i nadmiarowej. Wszystkie strony są wypełniane tylko w pewnej części (określonej współczynnikiem alfa), dzięki czemu przy wstawianiu rekordów, najpierw są one umieszczane w wolnej przestrzeni na stronie, a dopiero potem w przestrzeni nadmiarowej. Wraz ze zwiększeniem obszaru podstawowego wzrasta wielkość przestrzeni nadmiarowej (przestrzeń nadmiarowa rośnie proporcjonalnie do wzrostu wielkości obszaru podstawowego).
- Przeglądanie zawartości pliku danych zgodnie z kolejnością wartości klucza (*show file*) – wyświetlanie zaczynamy od pierwszego rekordu na stronie a kończymy na ostatnim rekordzie z przestrzeni podstawowej. Jeżeli wyświetlony element ma ustawiony wskaźnik to przenosimy się w odpowiednie miejsce i wyświetlamy rekord, na który wskazywał.
- Przeglądanie zawartości indeksu zgodnie z kolejnością wartości klucza (*show index*) – indeks posortowany jest według wartości klucza, więc wystarczy wyświetlić wszystkie indeksy po kolei. Podczas operacji wstawiania, usuwania, aktualizacji oraz reorganizacji wykorzystywany jest trzeci, tymczasowy plik, który po zakończeniu operacji zostaje usunięty.

Program wczytuje dane zarówno z klawiatury, jak i z pliku, do którego wcześniej należy podać ścieżkę. Ograniczeniem jest to, że podczas uruchamiania programu i tworzenia pustej strony, dodawany jest do niej rekord o najmniejszym możliwym kluczu, przez co do przestrzeni podstawowej można dodać o jeden element mniej, niż wynikałoby to z wartości zmiennych.

3 Eksperymenty

Ze względu na brak konkretnych eksperymentów do przeprowadzenia, postanowiłem przeprowadzić po 50 operacji wstawiania, usuwania oraz aktualizacji rekordów. Gdy dodałem do bazy 50 kluczy, część z nich została zaktualizowana, a na koniec niektóre wartości zostały usunięte. Przykładowe query zostało zapisane do pliku załączonego razem z plikiem źródłowym kodu.

Następnym eksperymentem było wykonanie kolejnych 50 operacji wstawiania, a następnie 25 operacji usuwania i 25 operacji aktualizacji rekordów. Wszystkie testy zostały wygenerowane losowo w oparciu o poprzedni projekt.

4 Wyniki eksperymentów

Operacja	Średnia liczba odczytów	Średnia liczba zapisów
Eksperyment 1		
Wstawianie	13.21	10.34
Usuwanie	11.30	9.54
Aktualizacja	15.79	12.10
Eksperyment 2		
Dowolna	12.58	10.20

5 Wnioski

We wszystkich eksperymentach przyjąłem, że:

- rozmiar strony w przestrzeni podstawowej wynosi 10,
- rozmiar bufora odczytu/zapisu zarówno pliku danych, jak i pliku z indeksami, wynosi 10,
- współczynnik alfa wynosi 0.5,
- ilość rekordów w przestrzeni przepełnienia stanowi jest równa 1/5 maksymalnej ilości rekordów w przestrzeni podstawowej.

Eksperyment 1: We wszystkich operacjach (wstawiania, aktualizacji i usuwania) najpierw należy odnaleźć odpowiednią stronę za pomocą indeksu, co wiąże się z takim samym kosztem. Następnie należy dodać lub usunąć rekord z odpowiedniej strony, po czym przepisać cały plik. Liczba operacji dyskowych podczas usuwania jest najmniejsza ponieważ wystarczy tylko przepisać plik, „po drodze" usuwając zadany rekord. Liczba odczytów i zapisów z/na dysk podczas wstawiania rekordu rośnie skokowo, ponieważ zwiększa się także ilość stron w przestrzeni podstawowej. W wynikach występują również pojedyncze wartość znacznie odbiegające od reszty wyników. Jest to spowodowane zapełnieniem przestrzeni nadmiarowej i uruchomieniem reorganizacji. Aktualizacja polega w pierwszej kolejności na usunięciu rekordu oraz na późniejszym wstawieniu nowych wartości do bazy. W związku z tym, iż jest to złożona operacji liczba operacji dyskowych potrzebna do jej wykonania jest najwyższa. We wszystkich przypadkach liczba odczytów przewyższała liczbę zapisów. Jest to związane między innymi z wyszukiwaniem klucza w indeksie.

Eksperyment 2: Zarówno liczba odczytów z dysku, jak i ilość zapisów na dysk uzyskana podczas działania programu jest niższa od wyników, których można by się było spodziewać na podstawie rezultatów pierwszego eksperymentu. W pierwszym eksperymencie najpierw wstawiliśmy, potem próbowaliśmy usunąć a na koniec aktualizować po 100 rekordów, natomiast w drugim teście operacje te były wykonywane na zmianę (w zależności od tego jak zostały wylosowane), dzięki czemu usuwanie rekordu, zwalniało miejsce na nowy wpis co skutkowało rzadszymi wywołaniami reorganizacji pliku.

6 Specyfikacja formatu pliku tekstowego

Plik testowy składa się z dowolnej liczby wierszy, z których każdy zawiera jedno polecenie. Polecenia mają różny format – najpierw podawany jest string, oznaczający operację do przeprowadzenia. Po znaku operacji mogą wystąpić wartości: napięcie i natężenie (rekord, na podstawie którego obliczany jest klucz) i/lub klucz. Dane w wierszu oddzielone są spacją.

Możliwe polecenia, wraz z formatem poleceń, zostały opisane w rozdziale 2 - "Realizowane operacje". Przykładowy format pliku wejściowego przedstawiam poniżej (plik *input_file.txt*):

```
add 10 20
add 32 32
add 43 83
add 432 32
add 83 312
add 23 81
add 1 5
add 5 3
add 43 12
add 84 32
add 23 54
add 65 54
show index
show file
reorg
show file
quit
```

Po wykonaniu tych poleceń dostajemy następujący output:

```
SHOW INDEX FILE
1 0
1024 1
13824 2
SHOW FILE
1 1 1
5 1 5
15 5 3
200 10 20
516 43 12
-1 0 0
-1 0 0
-1 0 0
-1 0 0
-1 0 0
1024 32 32
1242 23 54
```

1863 23 81
2688 84 32
3510 65 54
3569 43 83
-1 0 0
-1 0 0
-1 0 0
-1 0 0
13824 432 32
25896 83 312
1863 23 81
2688 84 32
3569 43 83
-1 0 0
-1 0 0
-1 0 0
-1 0 0
-1 0 0
Read = 4
REORGANIZATION
Reads = 10 Writes = 7
SHOW FILE
1 1 1
5 1 5
15 5 3
200 10 20
516 43 12
-1 0 0
-1 0 0
-1 0 0
-1 0 0
-1 0 0
-1 0 0
-1 0 0
-1 0 0
-1 0 0
-1 0 0
-1 0 0
-1 0 0
-1 0 0
-1 0 0
-1 0 0
1024 32 32
1242 23 54
1863 23 81
2688 84 32
3510 65 54

-1 0 0
-1 0 0
-1 0 0
-1 0 0
-1 0 0
3569 43 83
-1 0 0
-1 0 0
-1 0 0
-1 0 0
-1 0 0
-1 0 0
-1 0 0
-1 0 0
-1 0 0
13824 432 32
25896 83 312
1863 23 81
2688 84 32
3569 43 83
-1 0 0
-1 0 0
-1 0 0
-1 0 0
-1 0 0
-1 0 0
-1 0 0
-1 0 0
-1 0 0
-1 0 0
-1 0 0
-1 0 0
-1 0 0
-1 0 0
-1 0 0
-1 0 0
Read = 7