

Лекція 7. Алгоритми зовнішнього сортування

Сортування злиттям

Сортування простим злиттям

В основі даного методу внутрішнього сортування лежить процедура злиття двох упорядкованих масивів. Ці масиви повинні бути об'єднані таким чином, щоб вийшов один упорядкований масив.

Нехай є два масиви А і В, впорядковані за зростанням ключів. Злиття полягає в почерговому пересиланні елементів з масивів А і В у зону виводу С. Порядок пересилання в зону виводу залежить від результату попарного порівняння ключів масивів А і В.

Сортування злиттям провадиться на основі використання розглянутої вище процедури. Сутність сортування полягає в тому, що масив, який ми намагаємося відсортувати, розділяється на рівні групи елементів. Групи упорядковуються, а потім попарно зливаються, утворюючи нові групи.

На першому етапі кожна група містить два сусідніх елементи вихідного масиву. Елементи усередині груп упорядковуються (наприклад, методом вставки або включення). Потім відбувається попарне злиття груп. Кількість груп у списку зменшується доти, доки не буде отримана одна упорядкована група. Якщо число елементів непарне, то вводиться додатковий елемент із максимальним значенням. Після сортування він відкидається. Якщо число груп, сформованих на першому етапі, непарне, то непарна група переписується без злиття.

Нехай є 2 масиви, кожен з яких вже відсортований:

масив1	масив2	результат
1	2	1
3	6	2
5	7	3
8	10	5
9	14	6
11	15	7
17	19	8
		9
		10
		11
		14
		15
		17
		19

Злиття та зовнішнє сортування

Злиття

Зовнішнє сортування сортує файли, які не поміщаються цілком в оперативну пам'ять.

Зовнішнє сортування сильно відрізняється від внутрішнього. Справа в тому, що доступ до файлу є послідовним, а не паралельним як це було в масиві. І тому зчитувати файл можна тільки блоками й цей блок відсортувати в пам'яті й знову записати у файл.

Принципову можливість ефективно відсортувати файл, працюючи з його частинами й не виходити за межі частини забезпечує алгоритм злиття.

Під **злиттям** розуміється об'єднання двох (або більше) упорядкованих послідовностей в одну впорядковану послідовність за допомогою циклічного вибору елементів, доступних у цей момент.

Злиття - набагато більш проста операція, ніж сортування.

Ми розглянемо 2 алгоритми злиття:

1. Пряме злиття. Алгоритм Боуза - Нельсона
2. Природне (Неймановское) злиття.

Пряме злиття. Алгоритм Боуза - Нельсона

1. Послідовність a розбивається на дві половини b і c .
2. Послідовності b і c зливаються за допомогою об'єднання окремих елементів в упорядковані пари.
3. Отриманій послідовності привласнюється ім'я a , після чого повторюються кроки 1 і 2; при цьому впорядковані пари зливаються в упорядковані четвірки.
4. Попередні кроки повторюються, при цьому четвірки зливаються у вісімки й т.д., поки не буде впорядкована вся послідовність, тому що довжини послідовностей щоразу подвоюються.

Приклад

Вихідна послідовність

$A = 44\ 55\ 12\ 42\ 94\ 18\ 06\ 67$

- 1 $b = 44\ 55\ 12\ 42$
 $c = 94\ 18\ 06\ 67$
 $a = 44\ 94'\ 18\ 55'\ 06\ 12'\ 42\ 67$
- 2 $b = 44\ 94'\ 18\ 55'$
 $c = 06\ 12'\ 42\ 67$
 $a = 06\ 12\ 44\ 94'\ 18\ 42\ 55\ 67'$
- 3 $b = 06\ 12\ 44\ 94'$
 $c = 18\ 42\ 55\ 67'$
 $a = 06\ 12\ 18\ 42\ 44\ 55\ 67\ 94$

Операція, що однократно обробляє всю множину даних, називається фазою.

Найменший підпроцес, що, повторюючись, утворить процес сортування, називається проходом або етапом.

У нашому прикладі сортування здійснюється за три проходи. Кожний прохід складається з фази розбивки й фази злиття.

Головним мінусом сортування злиттям є подвоєння розміру пам'яті, спочатку зайнятої сортуємими даними. Розглянемо алгоритм із рекурсивним актом злиття, запропонований Боузом і Нельсоном і не потребуючим резерва пам'яті.

Він заснований на очевидній ідеї: злити дві рівні впорядковані частини можна злиттям їхніх початкових половин, злиттям кінцевих і злиттям 2-ї половини 1-го результату з 1-ю половиною 2-го результату, наприклад:

злиття початків та злиття кінців

1 7 8 11 + 4 5 6 9 =

злиття в середній частині

5 7 + 6 8

 =

Якщо частини не рівні або не діляться точно навпіл, процедуру уточнюють належним чином. Аналогічне злиття "половинок" можна звести до злиття "четвертушек", "восьмушок" і т.д.; має місце рекурсія.

```

Const n=200;
Type
  tipkl=word;
  tip = Record
    kl: tipkl;
    z:Array[1..50] of real
  End;

Var
  A: Array[1..n] of tip;
  j:word;

Procedure Bose (Var AA; voz:Boolean);
Var
  m,j:word; x:tip; {tip - тип сортируемых записів}
  A: Array [1..65520 div Sizeof(tip)] of tip Absolute AA;
  Procedure Sli(j,r,m: word); { r - відстань між початками
    частин, що зливаються, а m - їхній розмір, j - найменший
    номер запису
  }
  Begin if j+r<=n
  Then If m=1
  Then
  Begin If voz Xor (A[j].kl < A[j+r].kl) Then
  Begin
  x:=A[j];
  A[j]:= A[j+r];
  A[j+r]:=x
  End
  End
  Else
  Begin
  m:=m div 2;
  Sli(j,r,m); {Злиття "почав"}
  If j+r+m<=n Then
  Sli(j+m,r,m); {Злиття "кінців"}

```

```

Sli(j+m, r-m,m) End {Злиття в центральній частині}
End{блоку Sli};
Begin
m:=1;
Repeat
j:=1; {Цикл злиття списків рівного розміру: }
While j+m<=n do
Begin
Sli(j,m,m);
j:=j+m+m
End;
m:=m+m {Подвоєння розміру списку перед початком нового
проходу}
Until m >= n {Кінець циклу, що реалізує все дерево злиттів}
End{блоку Bose};
BEGIN
Randomize;
For j:=1 to n do
begin
A[j].kl:= Random(65535);
Write(A[j].kl:8);
end;
Readln;
Bose(A,true);
For j:=1 to n do
Write(A[j].kl:8);
Readln
END.

```

Природне (Нейманівське) злиття

Поєднуються впорядковані частини, що спонтанно виникли у вихідному масиві; вони можуть бути також наслідком попередньої обробки даних. Розраховувати на однаковий розмір частин, що зливаються, не доводиться.

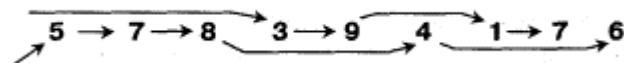
Записи, що йдуть у порядку неубування ключів, зчіплюються, утворюючи підсписок. Мінімальний підсписок один запис.

Приклад:

Нехай дані ключі записів

5 7 8 3 9 4 1 7 6

Шукаємо підписки



В один загальний список з'єднуються 1-й, 3-й, 5-й і т.д. підписки, в іншій - 2-й, 4-й і т.д. підписки.

Зробимо злиття 1 підписку 1 списку й 1 підписку 2 списку, 2 підписку 1 списку й 2 підписку 2 списку й т.д.

Будуть отримані наступні ланцюги

3 ---> 5 ---> 7 ---> 8 ---> 9 та 1 ---> 4 ---> 7

Підсписок, що складається із запису "6", пари не має й "примусово" поєднується з останнім ланцюгом, що приймає вид 1 ---> 4 ---> 6 ---> 7.

При нашому невеликому числі записів 2-й етап, на якому зливаються два

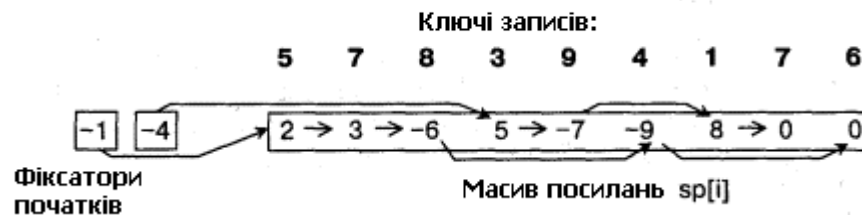
ланцюги, виявиться останнім.

У загальному випадку на кожному етапі підсписок - результат злиття початкових підписків 1 і 2 списку стає початком нового 1-го списку, а результат злиття наступних двох підписків - початком 2-го списку. Наступні утворені підписки по черзі включаються в 1-й і в 2-й список.

Для програмної реалізації створюють масив sp : елемент $sp[i]$ - це номер запису, який слідує за i -й.

Останній запис одного підписку посилається на перший запис іншого, а для розрізнення кінців підписків це посилання доповнюється знаком мінус.

Структура, що проявляє частини деякого початкового масиву:



```
Repeat {Повторення актів злиття підписків}
  If  $A[j].kl < A[i].kl$  Then {Вибирається менший запис}
    Begin  $sp[k] := j$ ;  $k := j$ ;  $j := sp[j]$ ;
    If  $j \leq 0$  Then {Зчеплення із залишком "i"-підписка}
      Begin  $sp[k] := i$ ; Repeat  $m := i$ ;  $i := sp[i]$  Until  $i \leq 0$ 
    End
  End
Else
  Begin  $sp[k] := i$ ;  $k := i$ ;  $i := sp[i]$ ;
  If  $i \leq 0$  Then {Зчеплення із залишком "j"-підписка}
    Begin  $sp[k] := j$ ; Repeat  $m := j$ ;  $j := sp[j]$  Until  $j \leq 0$ 
  End
End
End;
If  $j \leq 0$  Then Begin  $sp[m] := 0$ ;  $sp[p] := -sp[p]$ ;  $i := -i$ ;
 $j := -j$ ; If  $j < > 0$  Then  $p := r$ ;  $k := r$ ;  $r := m$  End
Until  $j = 0$ ;
```

У кінець сформованого підписка завжди заноситься нульове посилання ($sp[m] := 0$), тому що він може виявитися останнім.

Дія $sp[p] := -sp[p]$ позначає мінусом кінець раніше побудованого підписка.

У змінних i, j посилання на початки нових підписків, що зливаються - зі знаком мінус; його знімаємо. Перехід до нових підписків вимагає відновлення змінних p, k, r

Зовнішнє сортування

Зі світу "електронних" швидкостей перемістимося в світ менших швидкостей, світ з механічними переміщеннями й впливом інерції. Хоча диск і робить сотні обертів у секунду, проходять мілісекунди до моменту, коли потрібний запам'ятовувальний елемент виявиться під головкою читання/запису. Тому якщо будь-який відомий нам спосіб упорядкування в

пам'яті застосувати "механічно" до даних на диску, витрати часу виявляться, швидше за все, неприйнятними.

Сортування даних на магнітних дисках називають зовнішнім.

Складності при сортуванні на диску.

Досліджувати, а тим більш конструювати зовнішнє сортування неможливо, не представляючи, мало-мальски, специфіки дисків.

Диски мають запам'ятовувальні магнітні поверхні й обертаються з великою постійною швидкістю під головками читання/запису. Головка обслуговує одну поверхню.

При фіксованому положенні головок доступ відбувається до даних треків (окружностей), що перебувають під головками. Сукупність цих треків назвемо поточним циліндром.

Головки кріпляться на штанзі, подібної "коромислу" програвача. Після переміщення блоку головок (повороту штанги) поточним стає наступний циліндр. Циліндри нумеруються в порядку їхнього проходження блоком головок при русі штанги в одну сторону. Відстанню між елементами даних назвемо різницю номерів циліндрів, де вони розміщені.

Адресація (нумерація) елементів пам'яті на диску йде в межах циліндра з переходом по вичерпанні його обсягу на наступний, потім на третій і т.д. Файл даних записується один по одному адрес, але якщо вільної ділянки не вистачило, триває в іншій ділянці й, можливо, на іншому циліндрі (випадок фрагментації файлу).

Викладена досить спрощена модель диска, але її досить.

Хоча диски називають пам'яттю прямого доступу, час звертання до даних набагато більшого порядку, ніж час читання/запису в основній пам'яті, через

а) затримки, пов'язані з очікуванням моменту, коли потрібний елемент циліндра пройде під головкою;

б) переміщення головок, коли потрібні дані не з поточного циліндра.

Прогін - це проходження файлу в одному напрямку зі зчитуванням даних в пам'ять й, можливо, їхнім поверненням у файл.

{По можливості більшу частину роботи виконуйте з даними, викликаними в пам'ять, не звертаючись до диска до кінця обробки цієї частини.}

Стосовно до зовнішнього сортування це означає впорядкування можливо більших частин файлу внутрішнім сортуванням: ми заміняємо "швидкими" переміщеннями в пам'яті переміщення даних на диску. Зменшуючи число актів їхнього читання й перезаписи, ми одночасно виключаємо численні затримки "а", "б".

Збільшення фізичного розміру зчитувальної (записуваної) порції даних скорочує затримку й число актів читання (запису), а разом з ними й число проміжних рухів блоку головок, які назвемо ходами штанги або просто ходами.

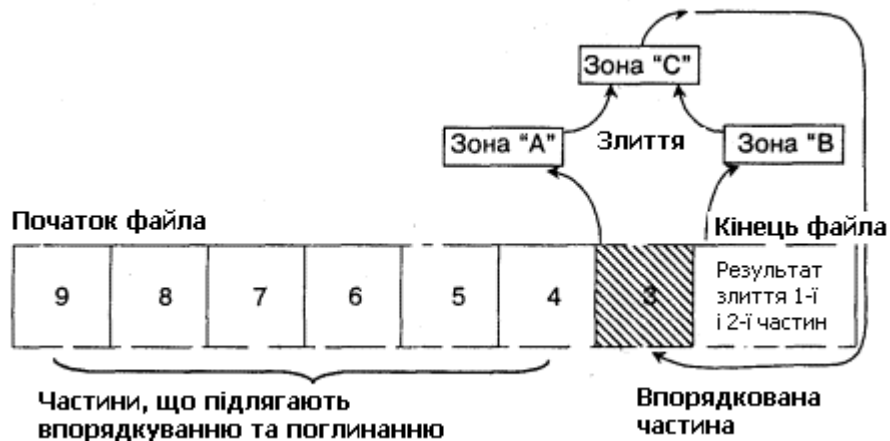
Час зовнішнього сортування залежить від

а) внутрішнього сортування частин файлу;

- б) багаторазового зчитування й запису даних на диск;
- в) ходів головки між актами зчитування/запису;
- г) дій у пам'яті при злитті впорядкованих частин.

Сортування методом поглинання.

Маючи кілька частин файлу й почавши зі злиття двох з них, будемо зливати всі наступні з більшою (зростаючою) упорядкованою частиною. Вона як би поглинає частину за частиною.



Перед поглинанням чергова частина файлу зчитується в зону "А" пам'яті, там упорядковується й залишається. Початок раніше впорядкованої частини зчитується в зону "В" і починається злиття, що переривається зчитуванням у зону "В", коли вона випорожнюється. У міру заповнення зони "С" записами акту злиття, уміст її переписується у файл (на місце поглинаємої частини і далі у бік кінця файлу). Якщо при злитті взяті всі записи поглинаємої частини, поглинання завершується передачею у файл із зони "С" залишку результату. Злиття також завершується, якщо вичерпано раніше впорядковану частину. Поглинання нею чергової частини відбулося.

Ходів у даному методі буде мало й під час сортування ви не почуєте характерного поскрипування механізму головок. Це дає економію часу й при невеликому розмірі файлу сортування проходить швидко.

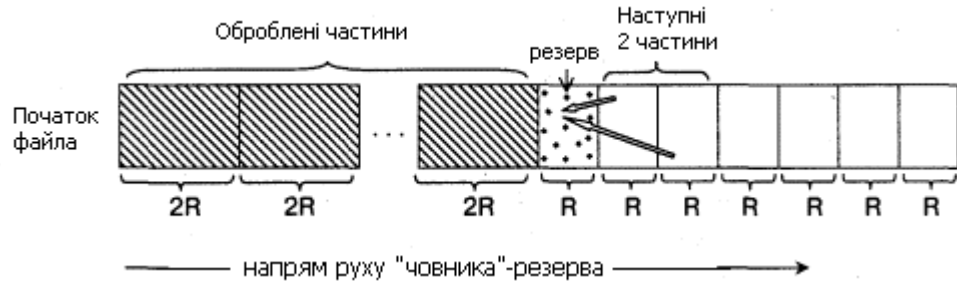
Човникове балансове злиття

На 1 етапі внутрішнього сортування частин у файлі створюють M упорядкованих частин, по можливості більшого розміру R . До них застосовують пряме злиття.

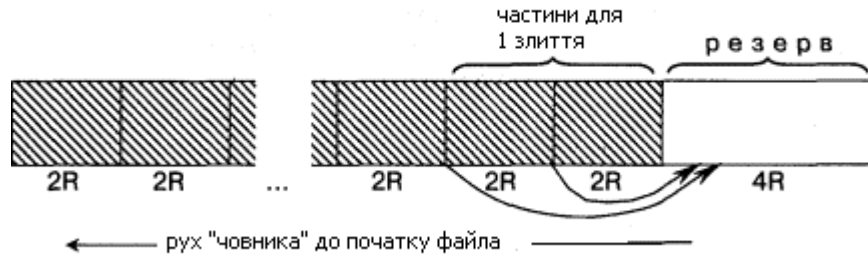
Створюють резервний дисковий простір файлу й розташовують його безпосередньо до або після частин, що зливаються, і переміщається до наступних частин по завершенні злиття. Його розмір не менше розміру R частини. Резерв і простір найближчої частини будуть заповнено результатом злиття двох перших частин, при цьому простір 2-ї частини звільниться й стане резервом, необхідним для злиття наступної пари частин і т.д.

У міру злиття частин резерв переміщається від початку файлу до кінця, потім назад і т.д., як човник. Оскільки місце результату не видалене від частин, що зливаються, ходи невеликі, поки самі частини невеликі. У процесі сортування "човник" збільшують, тому що ростуть частини:

а) злиття частин розміром R (проміжна ситуація)



б) Наступний "прохід" човника злиття частин розміром $2R$



Стрілками показане переміщення даних при злитті.

Помітимо, що резерв можна збільшувати, коли він наприкінці файлу; це відбувається один раз за 2 прогони човника.

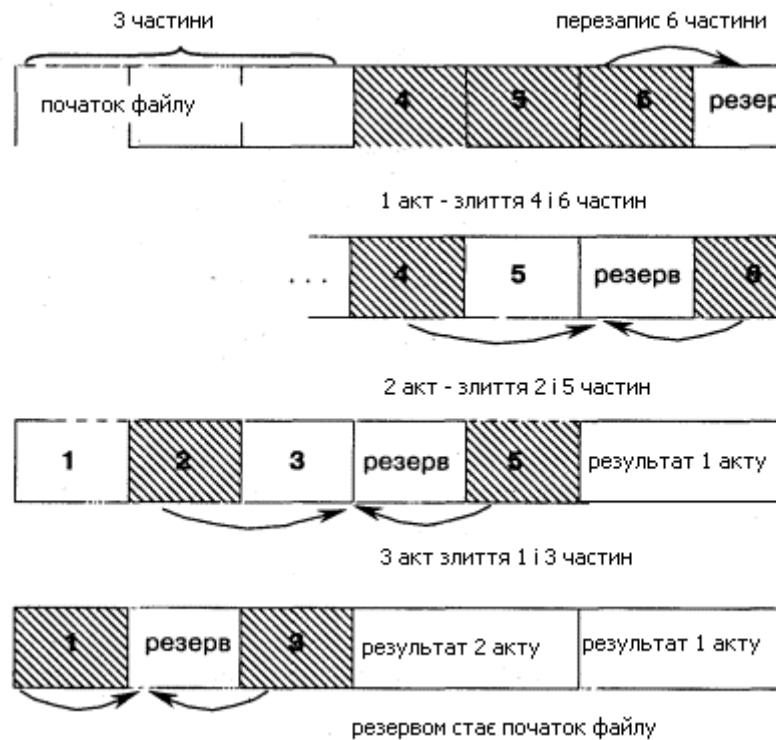
Усе було б добре, однак, човник завбільшки половини файлу може займати його початок.

Добре б обмежити ріст розміру "човника", тому що цей ріст збільшує розмір ходів.

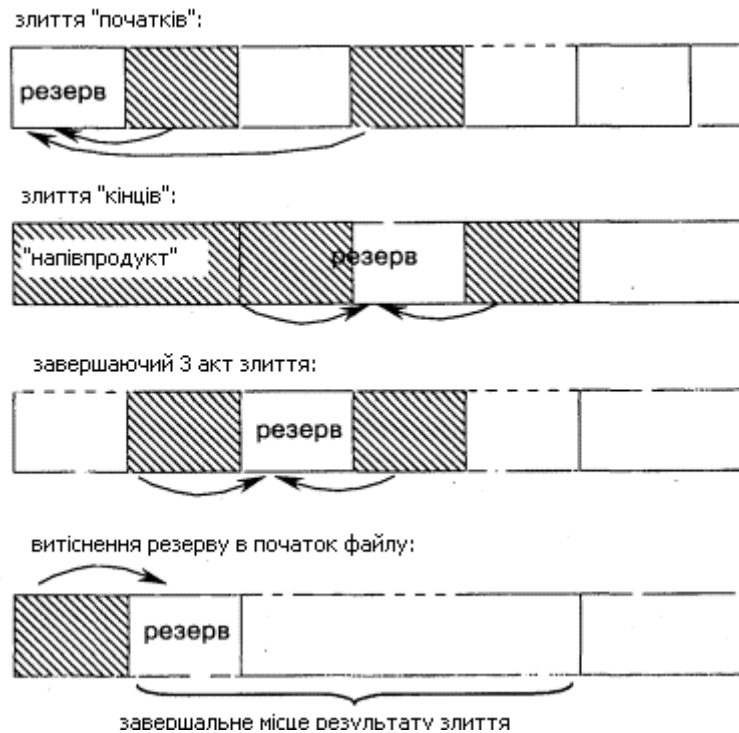
Шляхом модифікації кінцевих етапів злиття доможемося, щоб цей ріст зупинився на розмірі $D = 1/6$ розміру файлу.

Для цього однак треба, щоб програма так визначила вихідні розмір і положення резерву (на початку або наприкінці файлу), щоб у момент, коли залишаться всього 3 великі частини, резерв, що має розмір D , стояв на початку файлу.

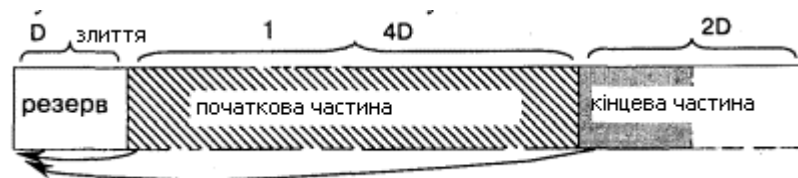
в) етап злиття (з підетапами), коли у файлі 6 частин.



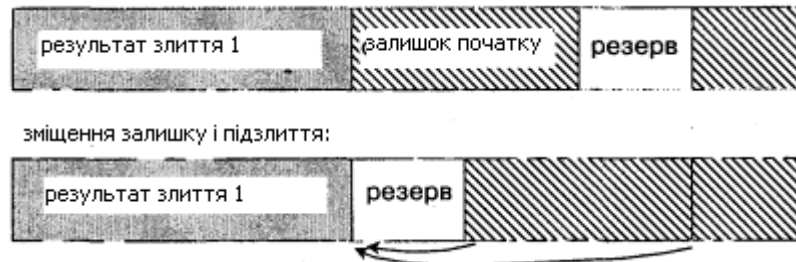
г) етап злиття "по половинах "(природне злиття), коли частин 3. Кінцева частина файлу на цьому етапі не використовується.



д) заключний етап злиття; спочатку беремо для злиття першу половину кінцевої частини й всю початкову частину:



По закінченні злиття 1 з'ясовується, чи треба змістити залишок початкової частини, щоб звільнити місце для кінця результату й продовжити злиття ("підзлиття"). Підзлиття не потрібне, якщо в ході злиття початкова частина вичерпана, треба лише витиснути резерв у кінець файлу, переписавши закінчення.



І при злитті 1, і при підзлитті розмір резерву не менше мінімально необхідного.