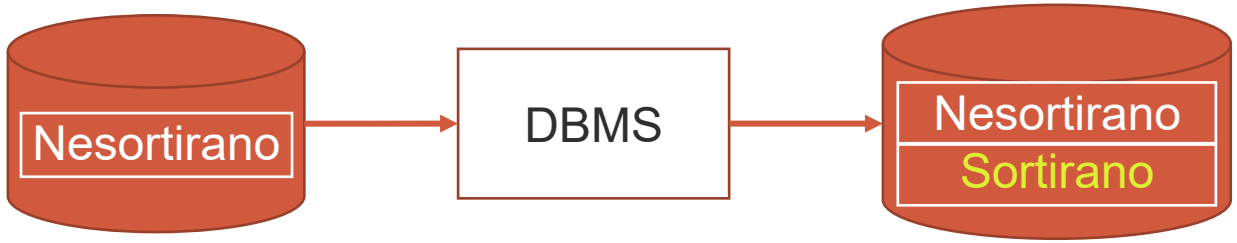
Particionisanje

Zašto particionisanje?

- Postoje operacije koje zahtevaju uređene ili grupisane podatke na osnovu vrednosti nekog polja. Podaci u tabelama i međurezultatima (koji se generišu tokom izvršavanja upita) često nisu organizovani na odgovarajući način pa je potrebno u letu kreirati privremene strukture.
  - **Indirektni zahtevi (zahtevano algoritmom neke operacije)**
    - DISTINCT
    - GROUP BY
    - Neke vrste JOIN algoritama (sort-merge)
  - **Eksplicitni zahtevi za uređivanjem**
    - ORDER BY
    - Kreiranje privremenih indeksa nad neuređenim slogovima (bulk-loading tree indexes)
- Definisanje algoritma za implementaciju operacija relacione algebre mora da uključi mogućnost da podaci na koje se primenjuju prevazilaze veličinu raspoloživog RAM-a

Sortiranje i heširanje

- Ulazni fajl F:
  - Koji sadrži slogove relacije R
  - Zauzima N blokova



- U RAM-u postoji prostor za fiksan broj blokova, B

- **Sortiranje**
  - Daje izlazni fajl F_s koji sadrži sortirane torke polazne relacije
- **Heširanje**
  - Daje izlazni fajl F_h u kom su slogovi sa istom heš vrednošću spakovani jedan za drugim.

Sortiranje

External-merge sort

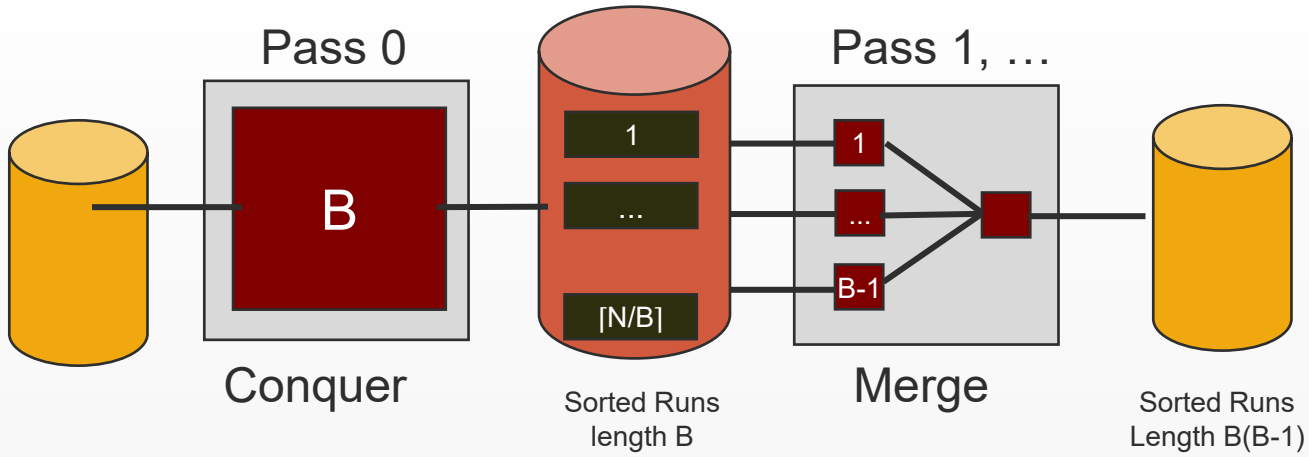
Sortiranje podatka koji prevazilaze veličinu raspoloživog RAM-a

Sortiranje

- Kako sortirati ako je tabela veća od raspoloživog RAM prostora?
- Relaciji se može pristupati kao sortiranoj ako izgradi indeks nad poljem za sortiranje, nakon čega bi se taj indeks koristio za čitanje relacije u sortiranom redosledu.
  - Tako sortirana relacija je sortirana *logički*, a ne *fizički*.
  - U tom slučaju čitanje torki može zahtevati pristup disku za svaki pojedinačni zapis, što može biti skupo, s obzirom na to da broj zapisa može biti znatno veći od broja blokova.
  - Zbog toga se često smatra poželjnim da zapisi budu fizički poređani na disku.
- **Spoljašnje sortiranje** - sortiranje relacije (fizičko) koja prevazilazi kapacitet raspoložive radne memorije. Najpoznatija tehnika spoljašnjeg sortiranja je
  - *External sort-merge* – sortiranje spajanjem sortiranih, ili samo sortiranje spajanjem

External Sort-Merge – osnovna ideja

- Osnovna ideja
  - Prva faza (faza parcijalnog sortiranja) – tabelu podeliti na porcije (*runs*) i sortirati pojedinačno svaki
  - Druga faza (spajanje u više iteracija)- sortirane porcije spajati u veće sortirane porcije, sve dok rezultat spajanja ne bude cela tabela.
- N – broj strana tabele, B – broj raspoloživih strana u baferu, *run* (porcija) – sortirani međurezultat



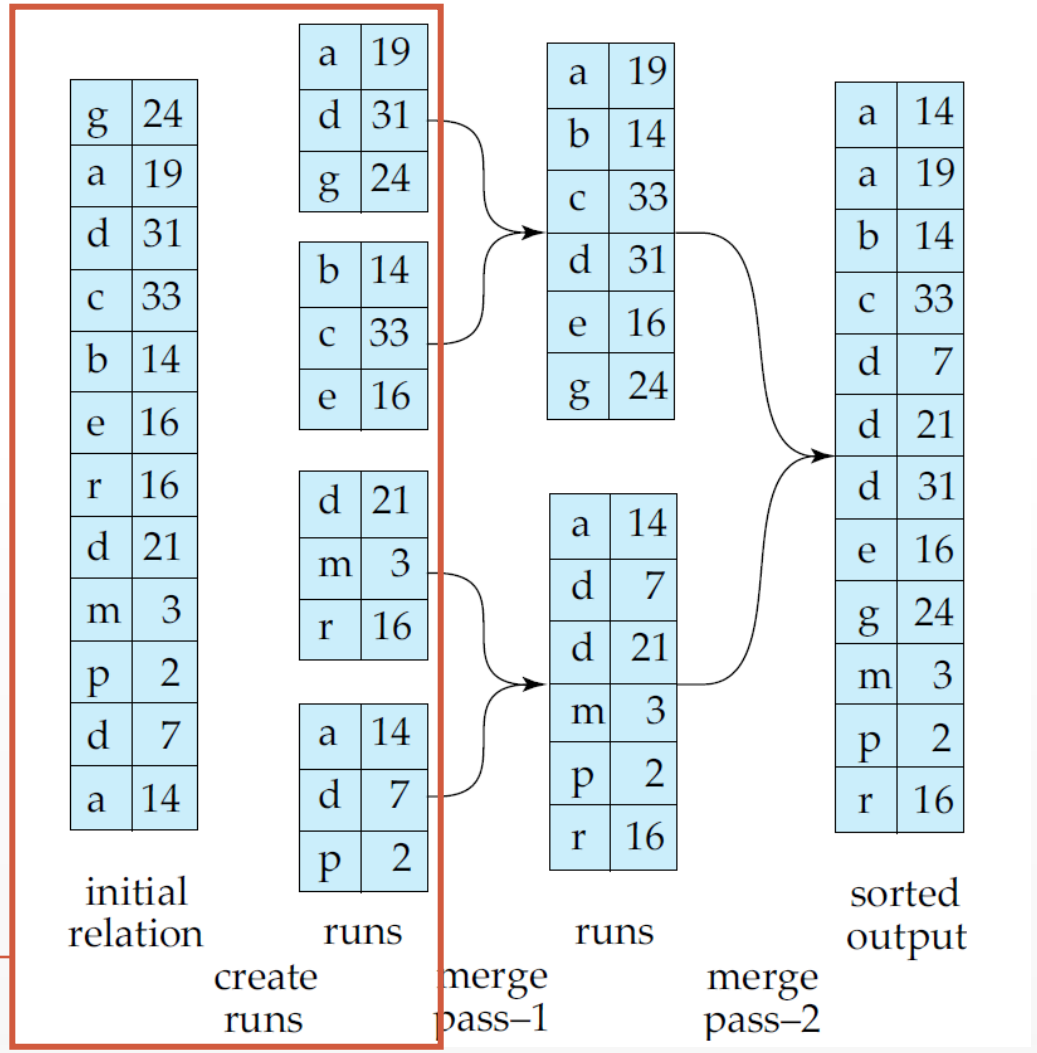
NAPOMENA
Algoritam koji je opisan ovde se u literature naziva i *General external sort-merge*, jer postoji i tzv. naivna verzija u kojoj je B = 3, pri čemu se u prvoj fazi sortira strana po strana i koristi samo 1 bafer strana.

External Sort-Merge – opis algoritma

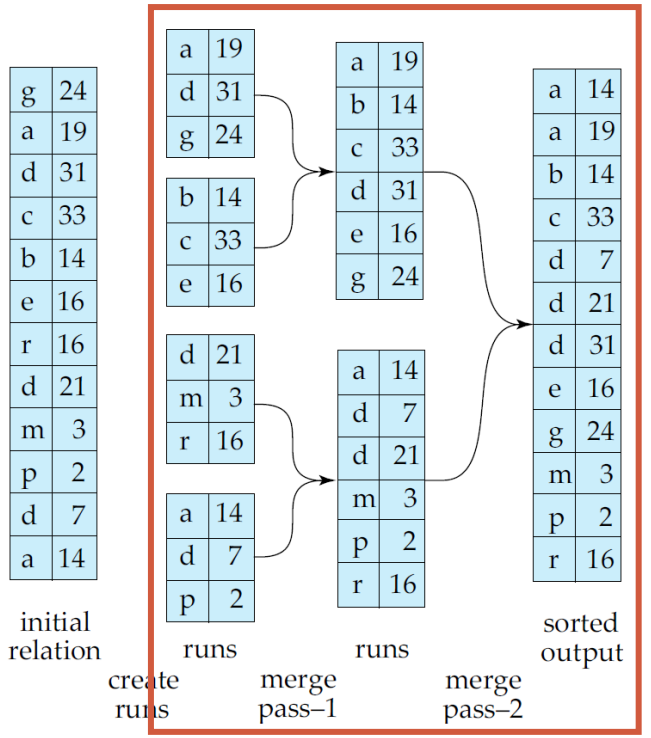
N – broj strana tabele, B – broj raspoloživih strana u baferu, pretpostavka da je $B < N$

1. Faza sortiranja (Nulti prolaz kroz tabelu) – kreiranje sortiranih porcija

   Učitavanje po B strana iz tabele, sortiranje i upis na disk (fajl porcije R_i).
   Svaka porcija R_i će imati po B strana, osim možda poslednje. Ukupno $\lceil N/B \rceil$ porcija



External Sort-Merge – opis algoritma (2)



2. Faza spajanja – spajanje sortiranih porcija, više prolaza

   Prolaz 1

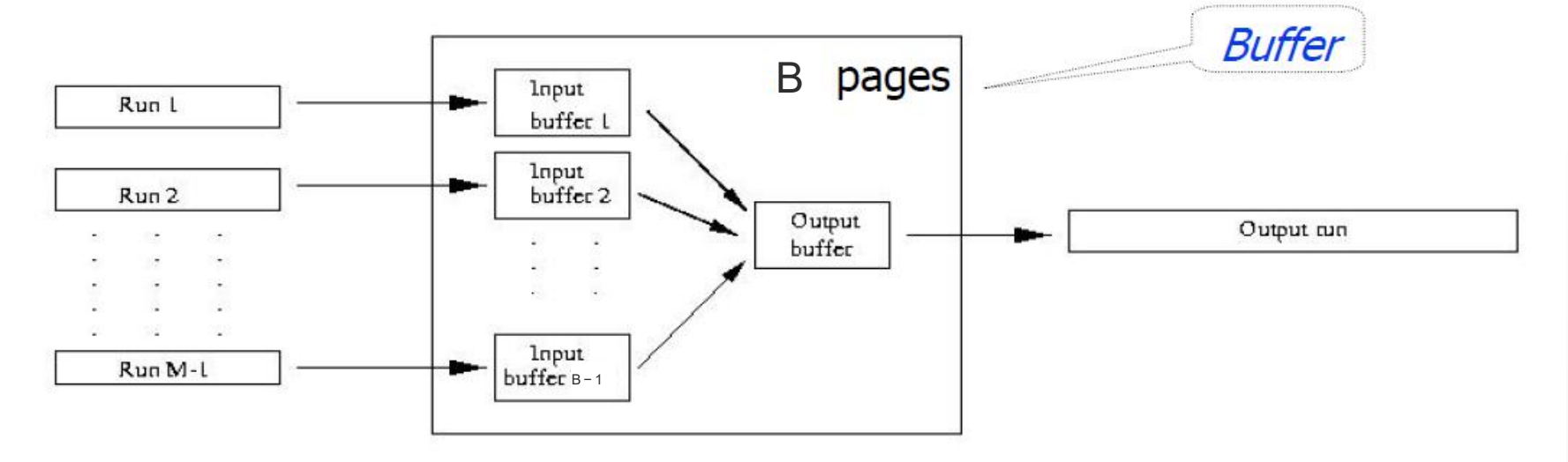
   1. Učitavanje po jedne (prve) strane iz svake od B-1 susedne porcije. B-1 strana bafera (ulazni bafer) se koristi sa učitavanje stranica porcija, a 1 ostavlja za sortirane slogove koja će biti upisivana na disk (izlazni bafer).
   2. Repeat sve dok svi ulazni baferi ne budu prazni
      1. Odabir prvog sloga (npr. slog sa najmanjom vrednošću ključa) od svih iz učitanih stranica i upis u izlaznu bafer stranu
      2. Ako je izlazni bafer pun prepisati ga na disk u novi R fajl.
      3. Ako je bafer blok bilo kog niza (Ri) prazan i nije kraj datoteke (end-of-file(Ri)), tada se učitava sledeći blok iz te datoteke u bafer blok.

   Broj sortiranih porcija je sada B puta manji, $\lceil \lceil N/B \rceil / B - 1 \rceil$.

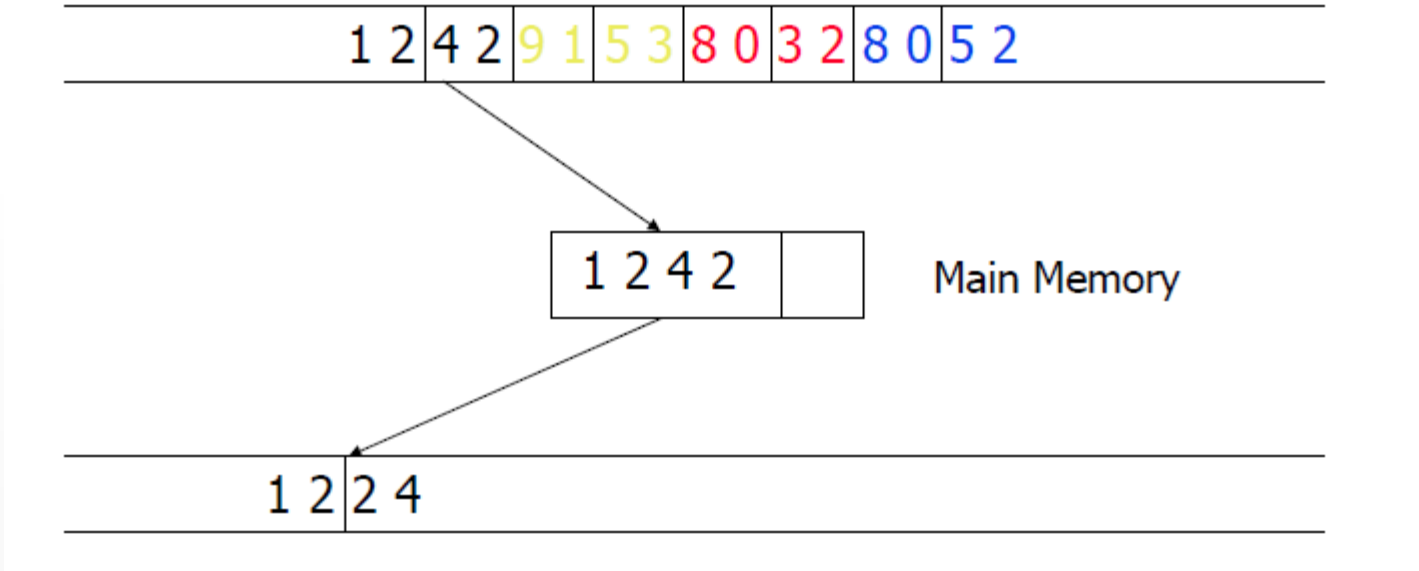
   Prolazi se ponavljaju sve dok se sve porcije ne spoje u jednu.

External Sort-Merge – algoritam (3)

- Jedan prolaz smanjuje broj porcija za faktor B − 1, i istovremeno stvara procije koje su B − 1 puta duže.

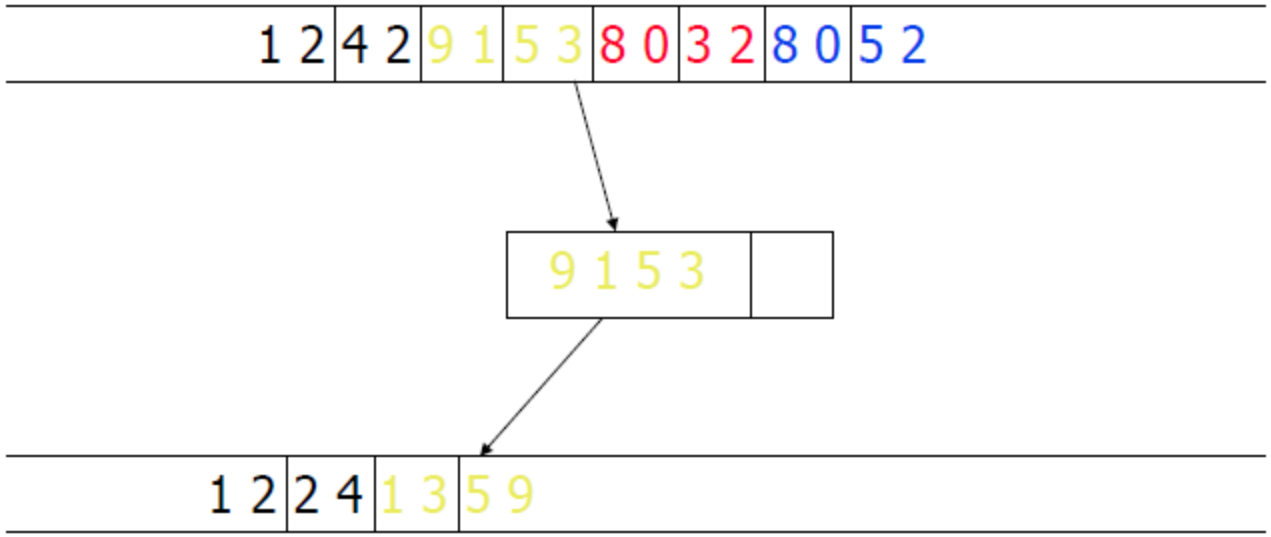


External Sort-Merge – primer (Prolaz 0)



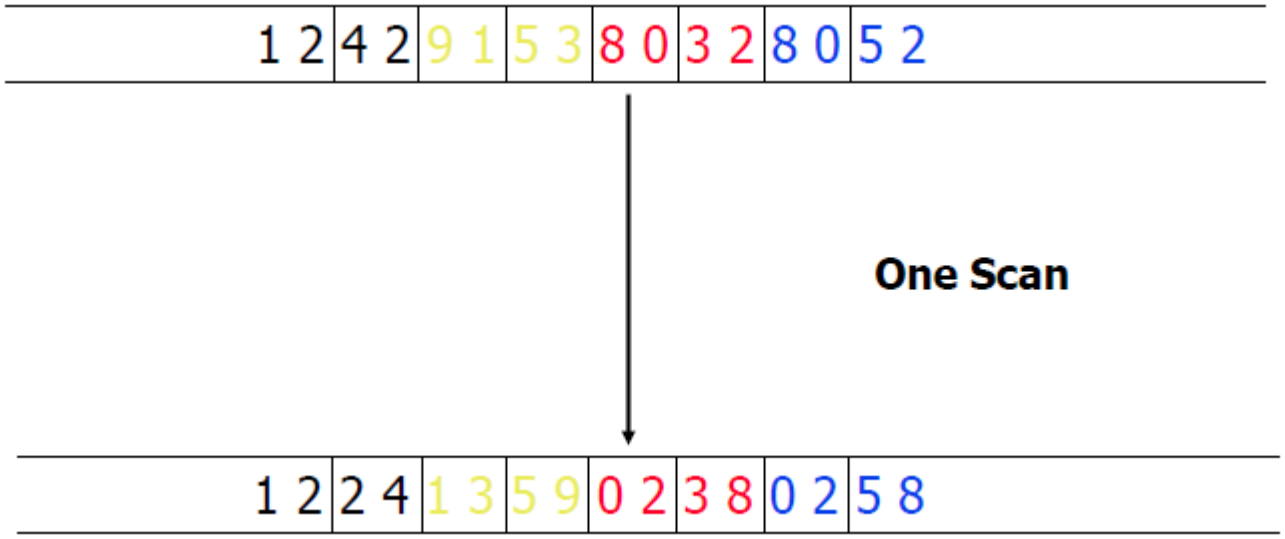
B = 3
N = 8
N/(B-1) = 4

External Sort-Merge – primer (Prolaz 0)



B = 3
N = 8
N/(B-1) = 4

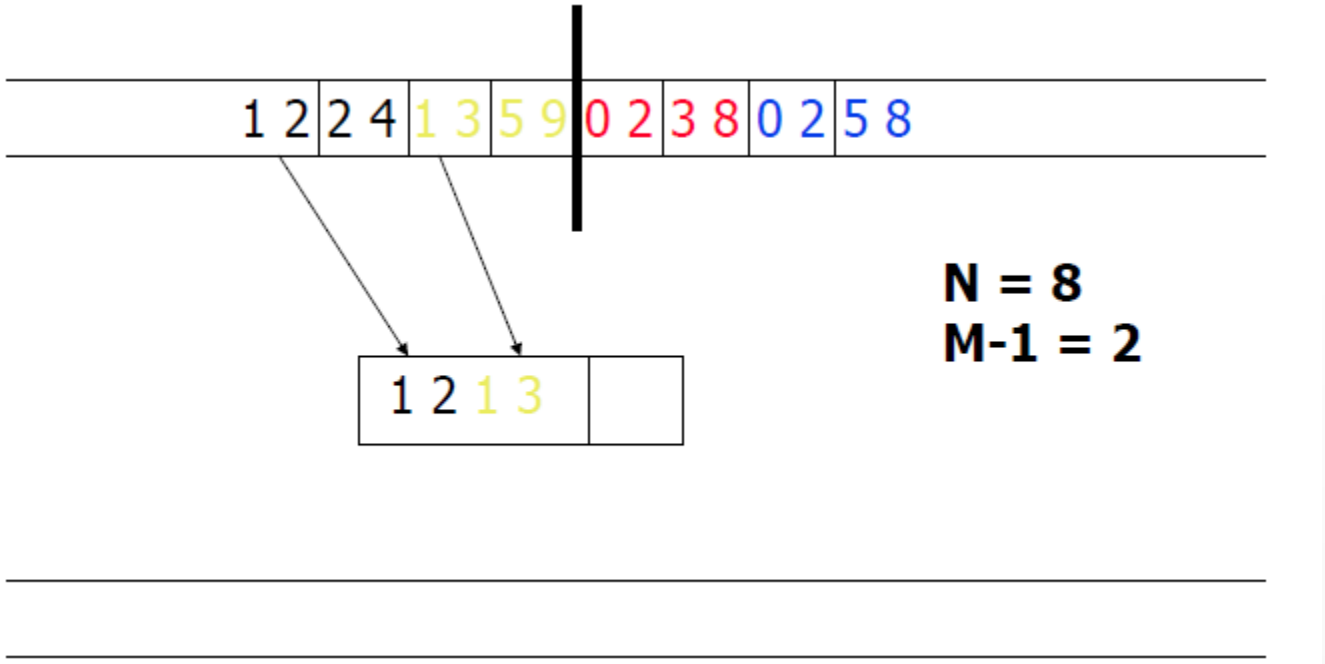
External Sort-Merge – primer (Prolaz 0)



B = 3
N = 8
N/(B-1) = 4

4 sortirane grupe sa po 2 strane

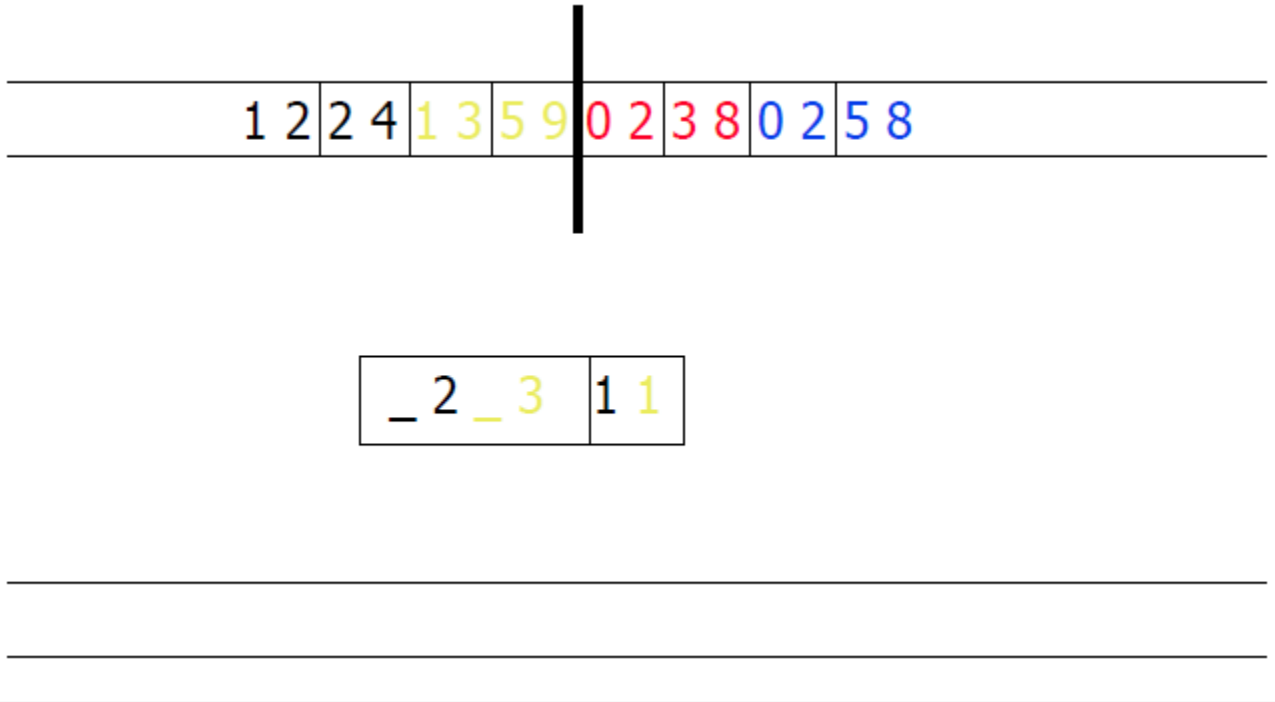
External Sort-Merge – primer (Prolaz 1)



B = 3
N = 8

4 ulazne porcije

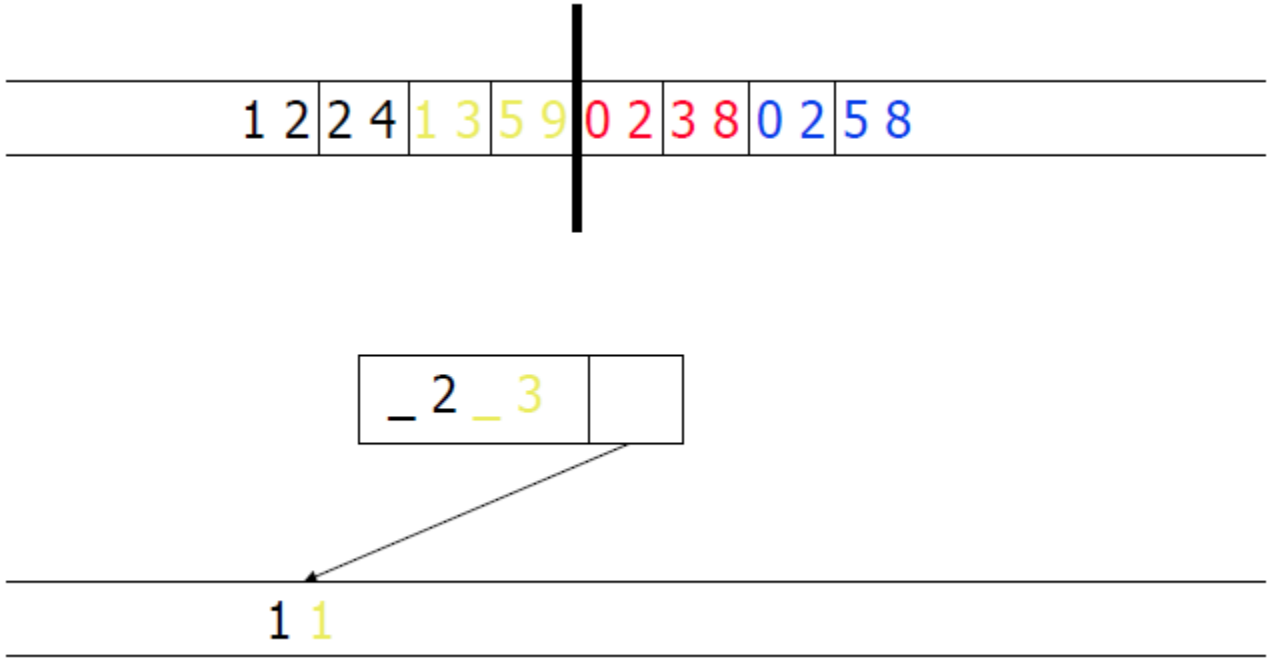
External Sort-Merge – primer (Prolaz 1)



B = 3
N = 8

4 ulazne porcije

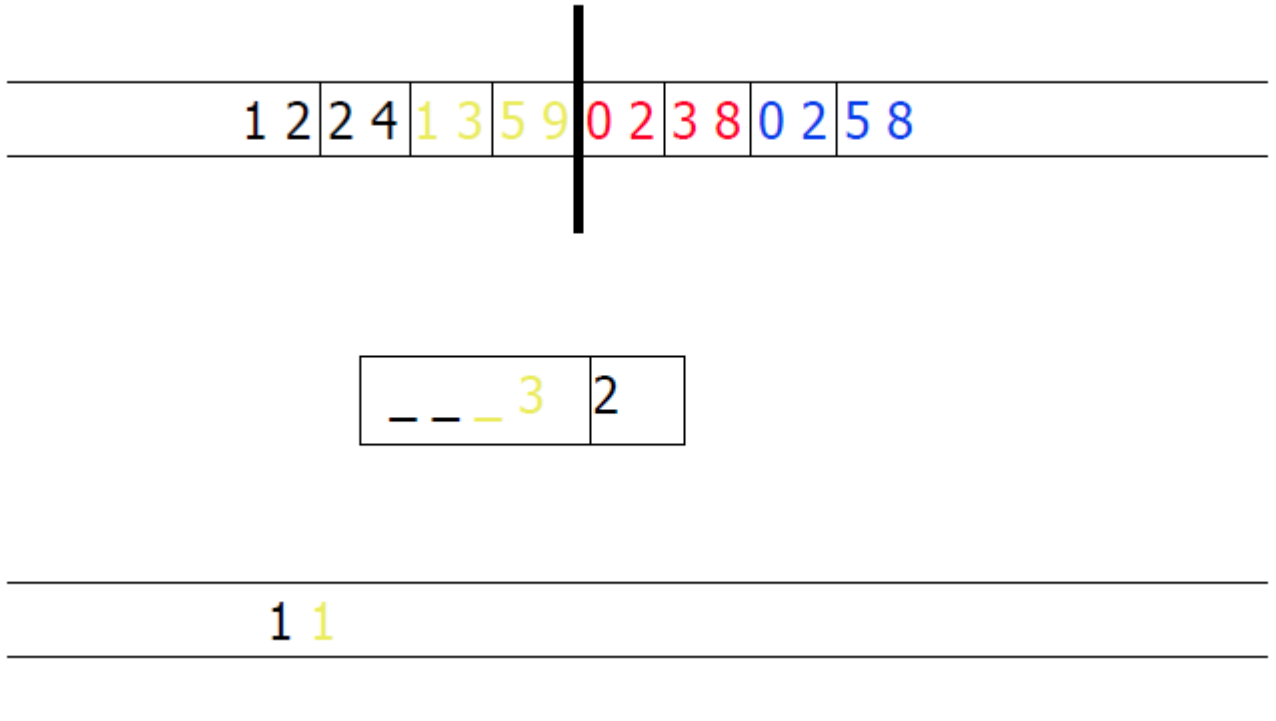
External Sort-Merge – primer (Prolaz 1)



B = 3
N = 8

4 ulazne porcije

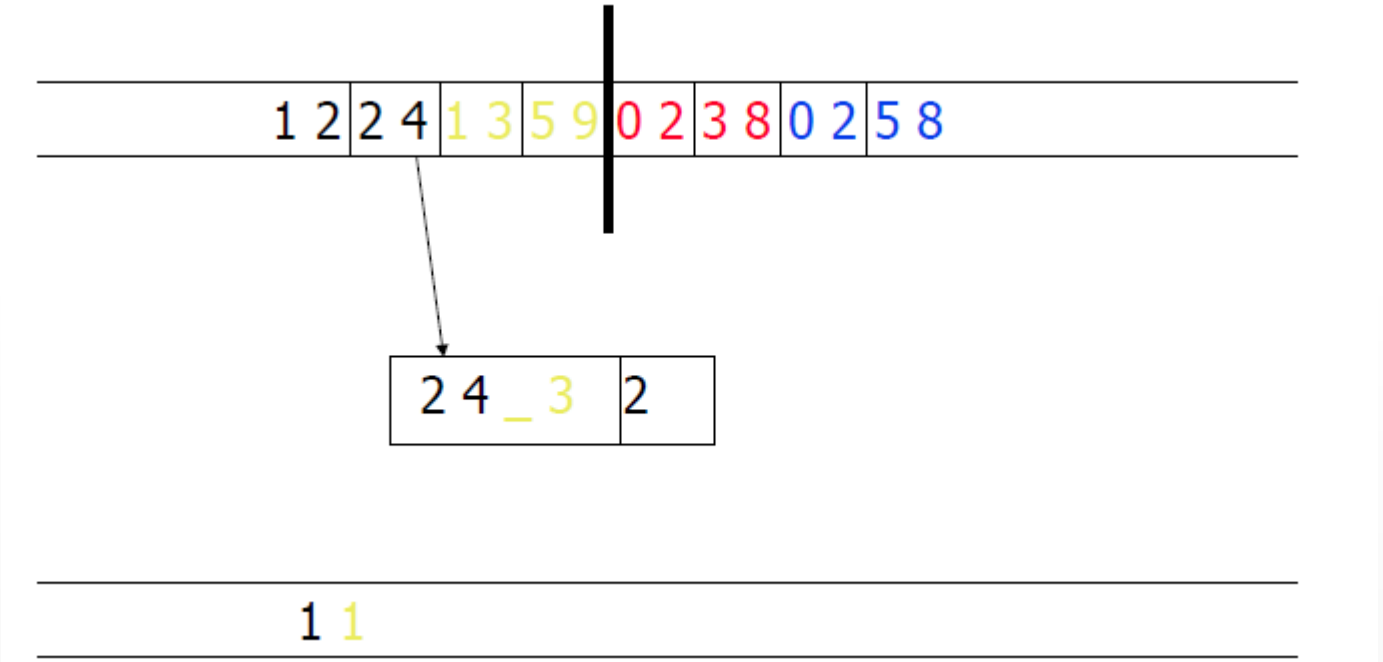
External Sort-Merge – primer (Prolaz 1)



B = 3
N = 8

4 ulazne porcije

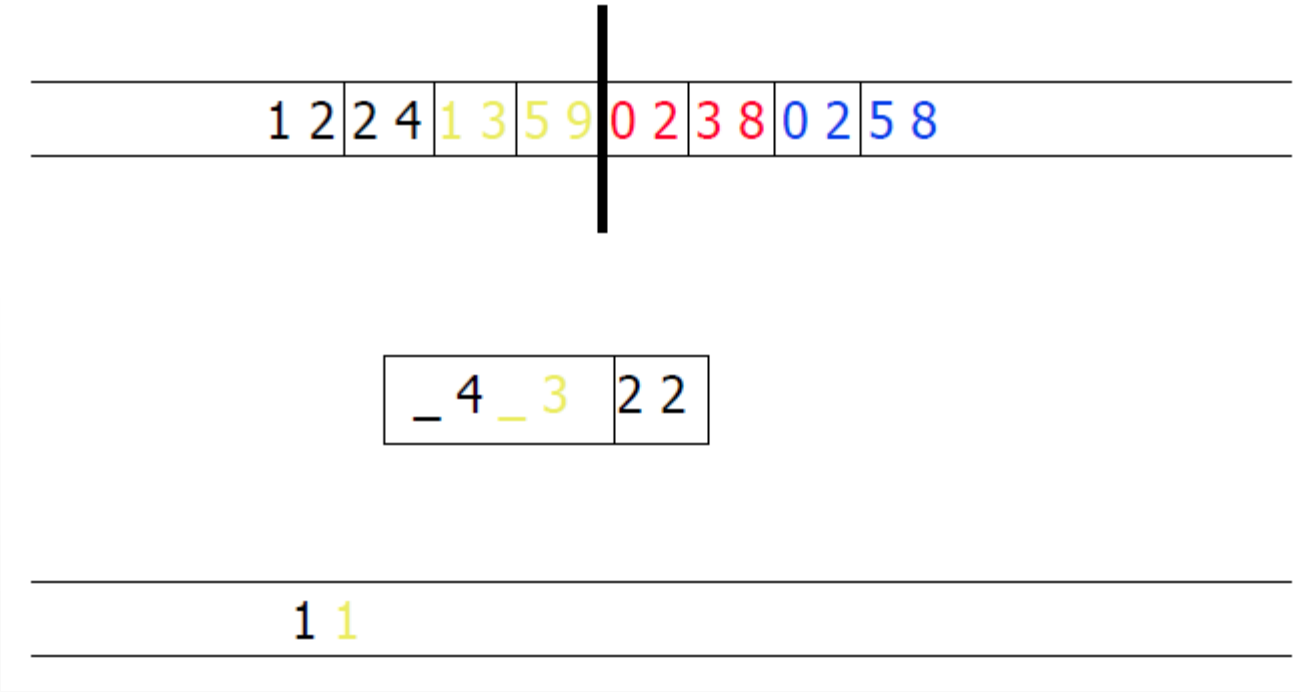
External Sort-Merge – primer (Prolaz 1)



B = 3
N = 8

4 ulazne porcije

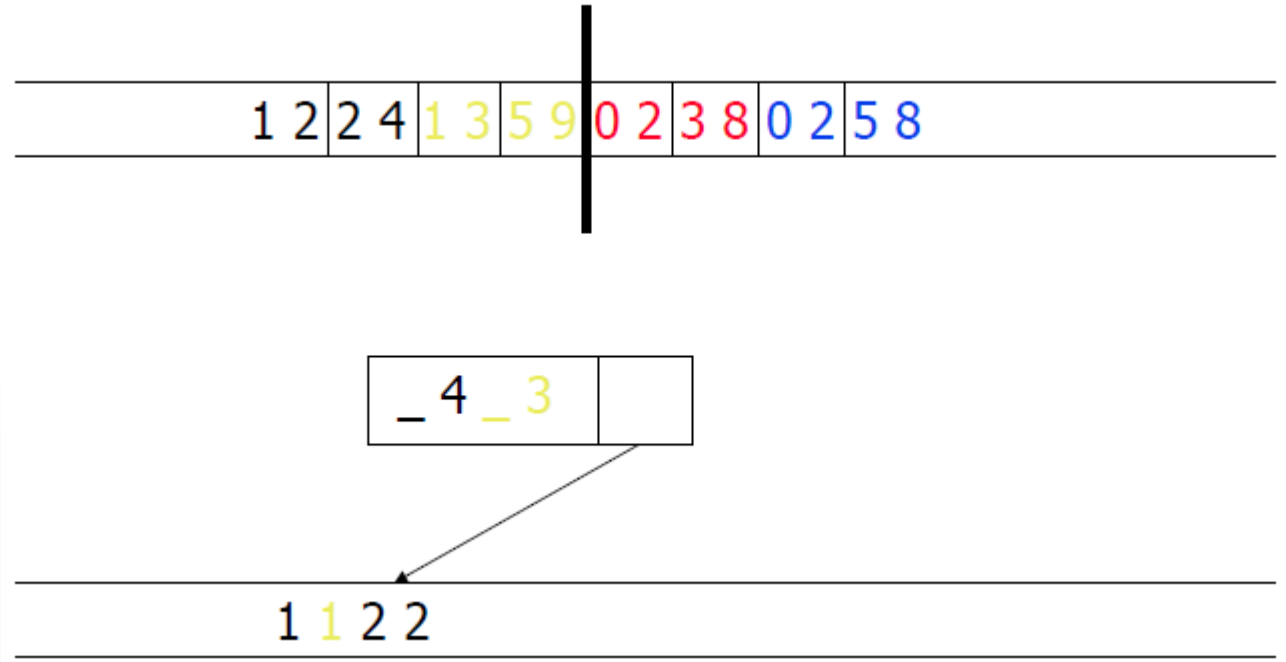
External Sort-Merge – primer (Prolaz 1)



B = 3
N = 8

4 ulazne porcije

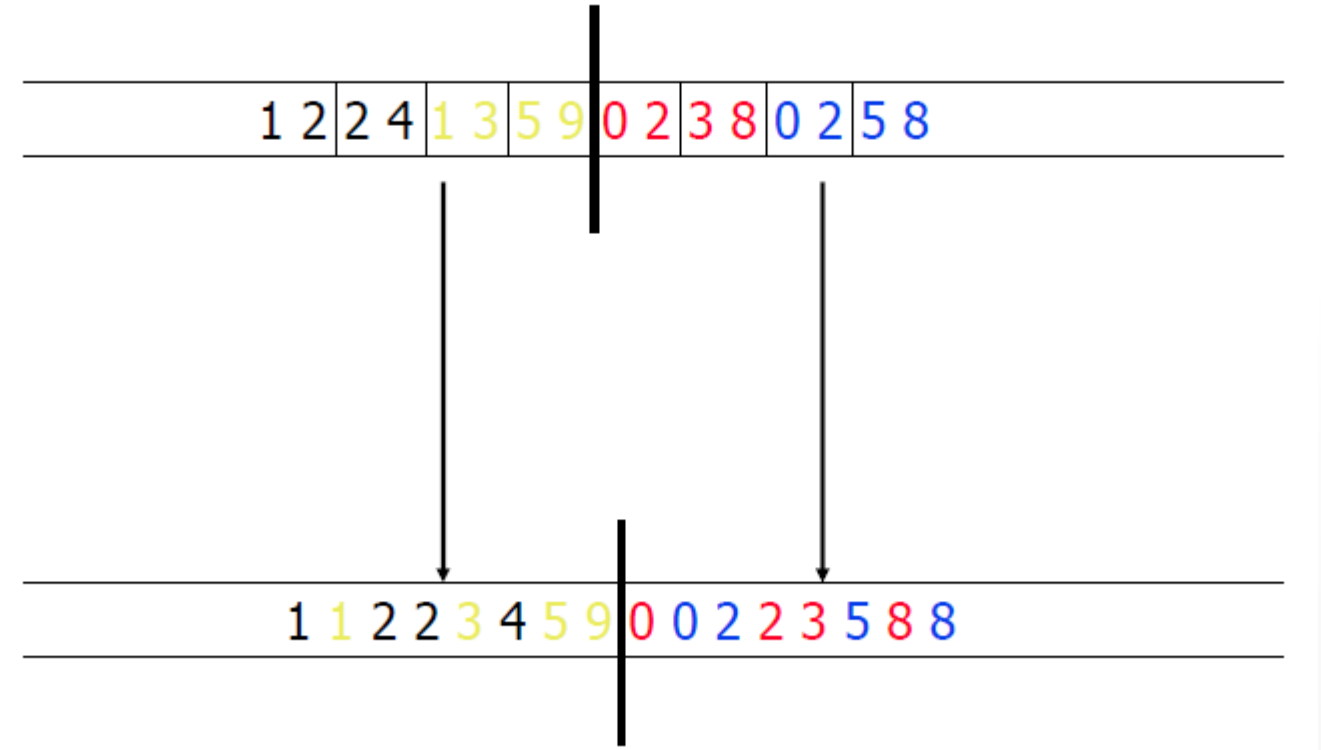
External Sort-Merge – primer (Prolaz 1)



B = 3
N = 8

4 ulazne porcije

External Sort-Merge – primer (Prolaz 1)



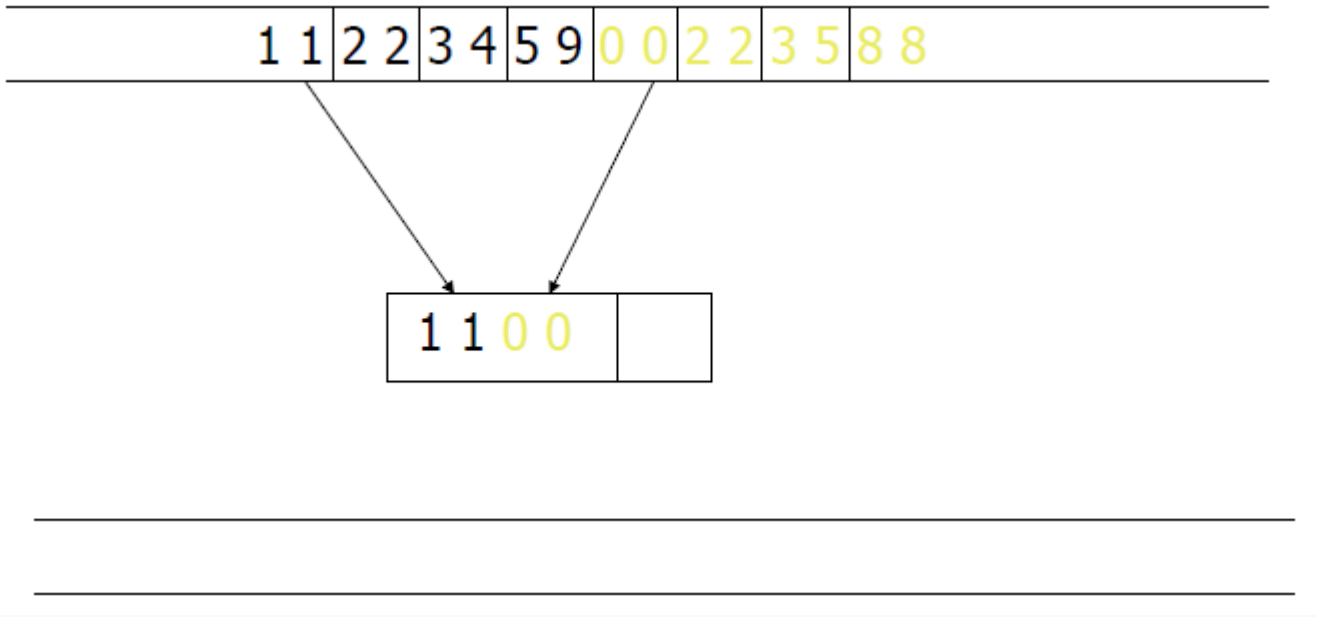
B = 3
N = 8

4 ulazne porcije

4/(B-1) = 2

2 sortirane grupe sa po 4 strane

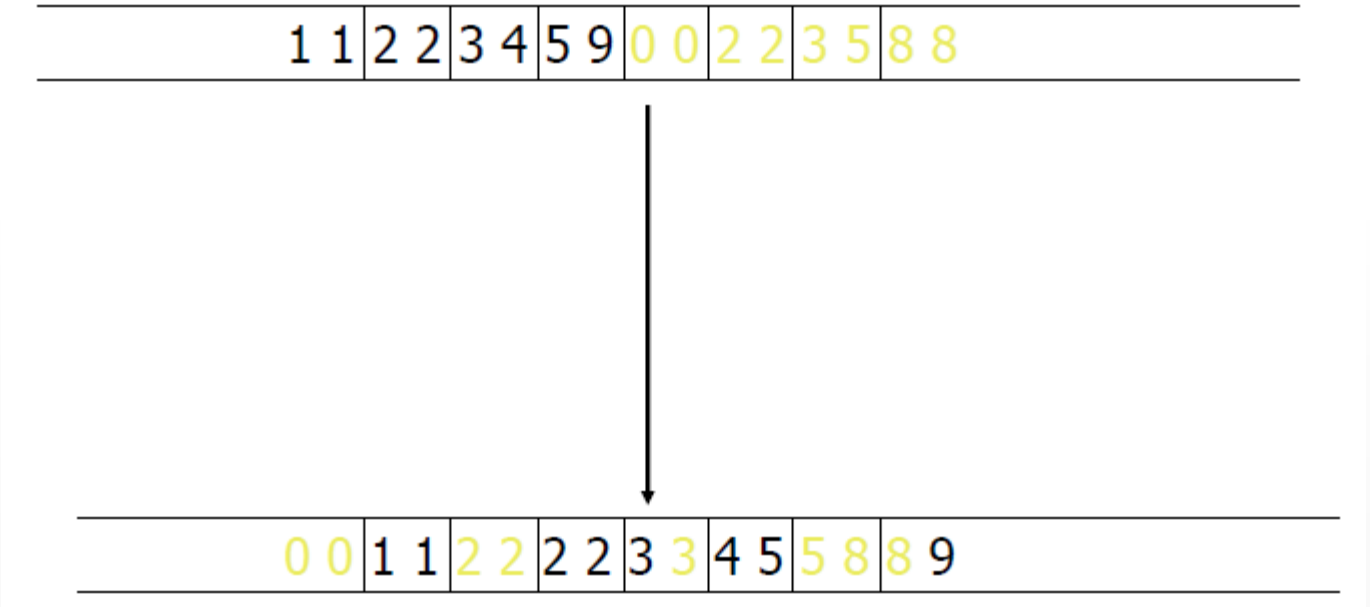
External Sort-Merge – primer (Prolaz 2)



B = 3
N = 8

2 ulazne porcije

External Sort-Merge – primer (Prolaz 2)



B = 3
N = 8

2 ulazne porcije

2/(B-1) = 1

1 sortirana grupa sa 8 strana

Troškovi spoljašnjeg sortiranja spajanjem

- Broj prolaza: $1 + \lceil \log_{B-1}[N/B] \rceil$
- I/O tošak – broj operacija nad stranama $2N * \#brojprolaza$

- Neka bafer ima 5 strana, a fajl sadrži 108 strana. Dakle, $B = 5, N = 108$
  Broj prolaza: $1 + \lceil \log_4 22 \rceil = 4$
  - Prolaz 0: 22 sortirane porcije sa po 5 strana, osim poslednje sa 3 strane
  - Prolaz 1: $\lceil 22/4 \rceil = 6$ sortiranih porcija sa po $4 * 5 = 20$ strana, osim poslednje sa 8
  - Prolaz 2: $\lceil 6/4 \rceil = 2$ sortirane porcije sa po 80, odnosno 28 strana
  - Prolaz 3: Sortirani fajl sa 108 strana

Troškovi – broj prolaza

N	B=3	B=5	B=9	B=17	B=129	B=257
100	7	4	3	2	1	1
1,000	10	5	4	3	2	2
10,000	13	7	5	4	2	2
100,000	17	9	6	5	3	3
1,000,000	20	10	7	5	3	3
10,000,000	23	12	8	6	4	3
100,000,000	26	14	9	7	4	4
1,000,000,000	30	15	10	8	5	4

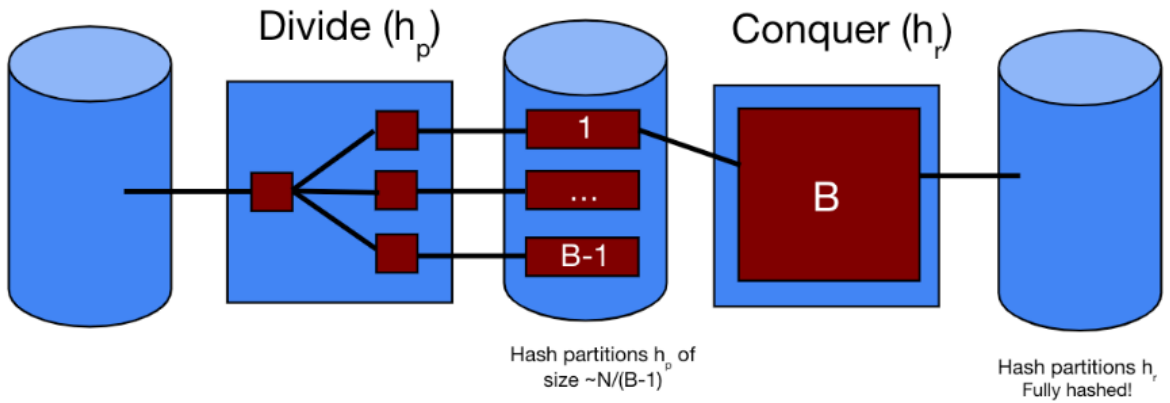
Heširanje

Tehnike particionisanja

Delimično uređivanje podatka koji prevazilaze veličinu raspoloživog RAM-a

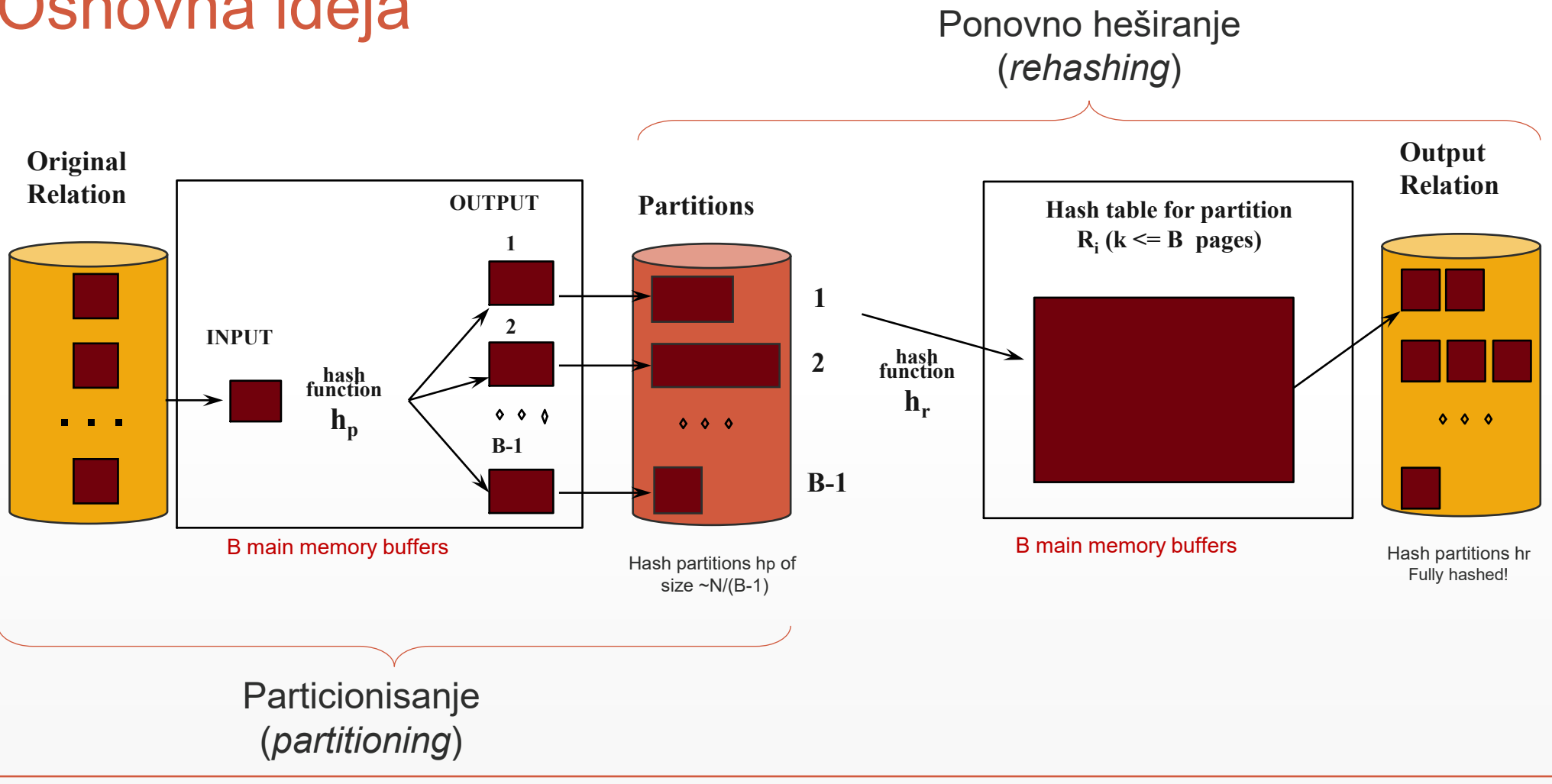
Motivacija i osnovna ideja

- Alternativa sortiranju u situacijama kada nije neophodno striktno sortiranje
  - Na primer, u slučaju uklanjanja duplikata ili grupisanja i agregiranja



- Osnovna ideja – Podeli-pa-vladaj
  - Faza 1 – Particionisanje - primenom jedne heš f-je tabela se deli na particije sa manjim brojem torki i različitih vrednosti ključa
  - Faza 2 – Ponovno heširanje – primena druge heš funkcije na svaku particiju pojedinačno
    formira se heš tabela sa torkama,
    popunjena heš tabela se upisuje na disk
    u njoj torke sa istim vrednostima ključa susedne.

Osnovna ideja



Troškovi spoljašnjeg heširanja

- Ukupan broj I/O operacija nad stranama je

$$2 * N * (\#brojprolaza) = 4 * N$$

- Opisani algoritam je odgovarajući ako ni jedna particija dobijena u prvoj fazi nema više od B strana. Koliko velika onda tabela može da bude?

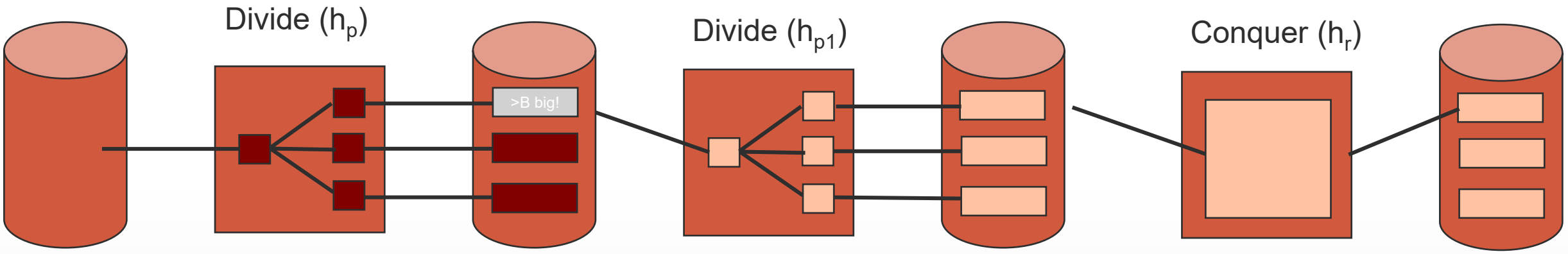
$$B * (B - 1)$$

U fazi 1 formira se B-1 particija. Svaka particija ima B strana.

- Ako tabela ima N strana, potreban je bafer od oko $\sqrt{N}$ strana, pod uslovom da su particije opterećene podjednako.

- Ako bafer nije dovoljno veliki, koristi se rekurzivno heširanje.

Rekurzivno heširanje



Tokovi I/O operacija

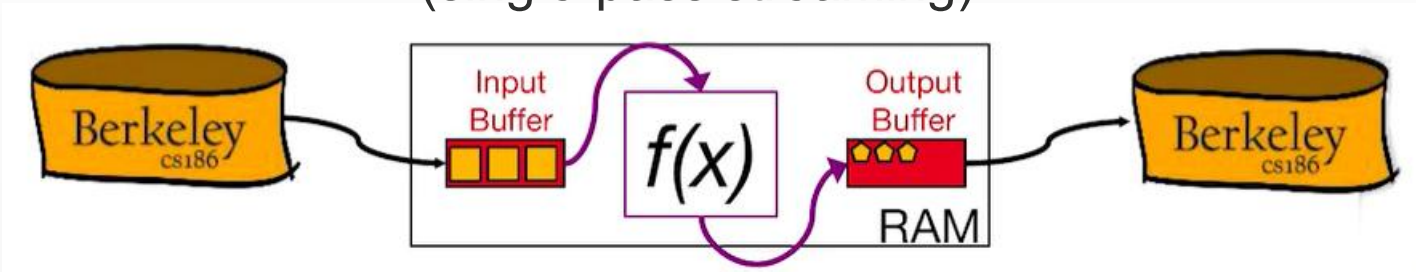
I/O streams

Optimizacija I/O tokova
Jednostruki i dvostruki tokovi

Jednostruki I/O bafer tok

- Jednostavan slučaj mapiranja – za svaku torku izračunaj izračunaj f(x) i upiši na disk
  Računa se na minimalnu upotrebu RAM-a i čitanje i pisanje u malo broju prolaza
  - I/O tok čine jedan ulazni i jedan izlazni bafer (sa po 1 stranom – generički slučaj)
  - U ulazni bafer se učitava jedna strana.
  - Slog po slog se obrađuju i razultati upisuju u izlazni bafer.
  - Kada se obrade sve torke iz ulaznog bafera učitava se nova strana.
  - Kada je izlazni bafer pun sadržaj se upisuje na disk, a sam bafer prazni.

Jednostruki bafer tok
(single-pass streaming)



Jednostruki I/O bafer tok

- Ulazni i izlazni baferi se ne pune/prazne sinhronizovano. Ako se puni ulazni bafer, jer ne ispražnjen, nije nužno da je istovremeno i izlazni pun.
- Ceo tok održava jedna nit.
  - To znači da ako je ulazni bafer ispražnjen da bi se nastavilo izračunavanje potrebno je sačekati da se završi operacija čitanja sa diska. Dakle sama nit je blokirana kad god se obavljaju I/O operacije
- Da li je moguće odvojiti zadatke sa I/O operacijama i zadatke obrade podataka koji su u memoriji?

  Da, uvođenjem dvostrukih tokova, tj. posebnih niti za ove dve vrste zadataka.

Dvostruki I/O bafer tok

- Dve niti koje rade paralelno
  - Glavna nit je zadužena za izračunavanje, vodi tok izračunavanja f(x) nad jednim parom (ulazni/izlazni) bafer
  - Druga nit je I/O nit i ima zadatak da učitava/prazni drugi par (ulazni/izlazni) bafera
- Dok glavna obrađuje podatke iz ulaznog bafera i upisuje izlazni, I/O nit puni svoj ulazni i prazni izlazni bafer
- Kada glavna nit završi sa obradom podataka iz svog ulaznog bafera, I/O nit i glavna nit razmenjuju parove bafera.

Dupli bafer tokovi
(double buffering)