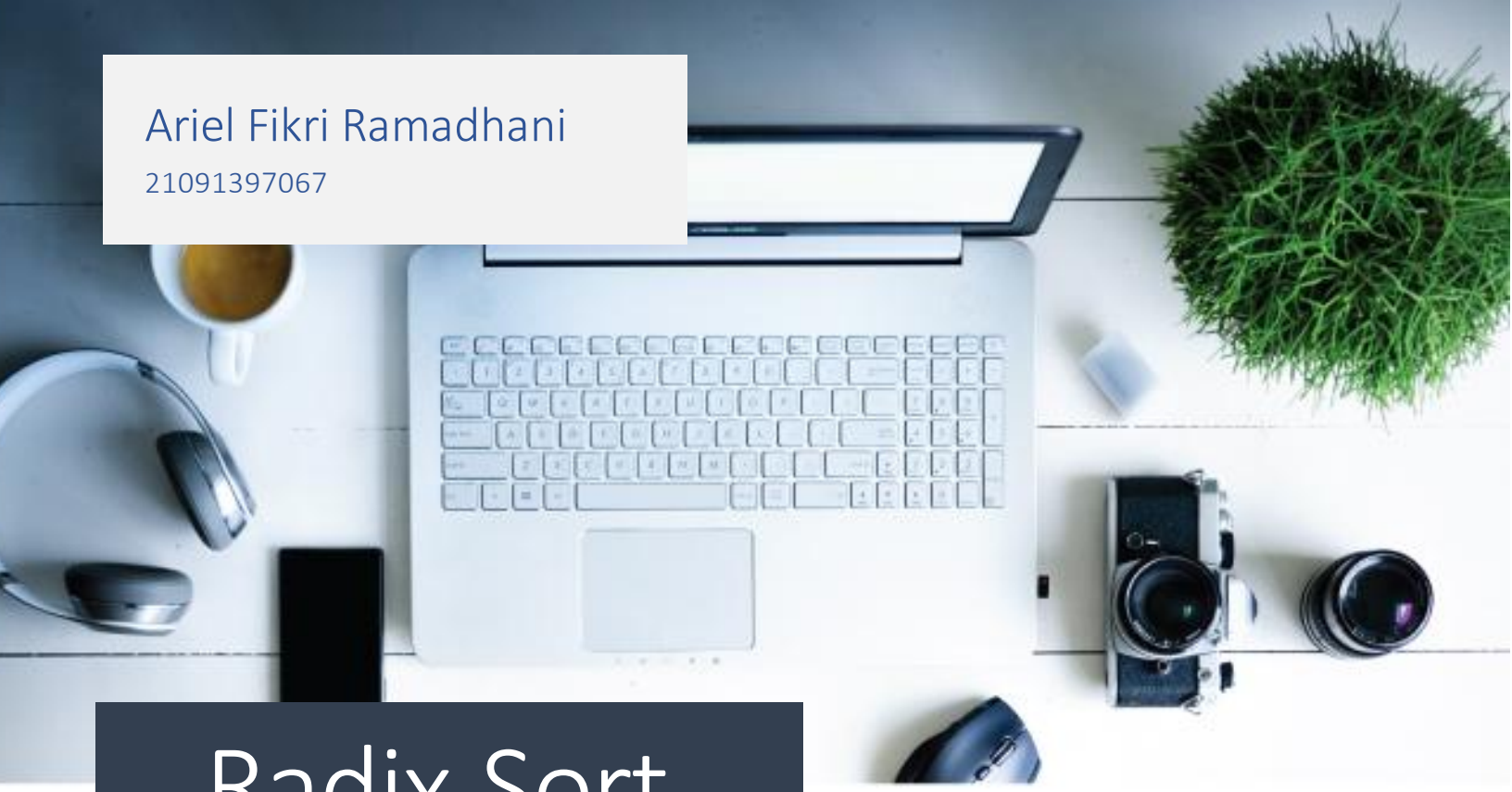
Ariel Fikri Ramadhani
21091397067



Radix Sort

Radix sort adalah algoritma stabil berbasis non-comparison yang menggunakan pengurutan penghitungan dengan sedikit modifikasi untuk mengurutkan bilangan dalam suatu array.

Kita dapat menggunakan counting sort hanya jika k linier terhadap n. Ini akan memakan waktu O(n2) ketika k meningkat menjadi n2, yang dimana kinerja yang diberikan lebih buruk daripada algoritma comparison-based lainnya seperti merge sort, dll.

Ini adalah salah satu kelemahan dari counting sort dan kesempatan untuk radix sort untuk memudahkan proses. Pengurutan radix dapat mengurutkan elemen untuk nilai k yang lebih besar dalam waktu linier.

Ide dari Radix Sort adalah melakukan pengurutan digit demi digit mulai dari digit yang paling rendah hingga digit paling akhir secara signifikan. Pengurutan Radix menggunakan pengurutan penghitungan sebagai subrutin untuk sort.

Pengaplikasian Radix Sort:

Di komputer biasa, yang merupakan mesin akses acak sekuensial, di mana catatan dikunci oleh beberapa bidang radix sort digunakan. Misalnya, Anda ingin mengurutkan tiga tombol bulan, hari dan tahun. Anda dapat membandingkan dua catatan pada tahun, kemudian pada seri pada bulan dan akhirnya pada tanggal. Atau, pengurutan data tiga kali menggunakan pengurutan Radix pertama pada tanggal, kemudian pada bulan, dan terakhir pada tahun dapat digunakan.

Itu digunakan di mesin sortir kartu yang memiliki 80 kolom, dan di setiap kolom, mesin hanya bisa melubangi 12 tempat. Penyortir kemudian diprogram untuk menyortir kartu, tergantung pada tempat kartu telah dilubangi. Ini kemudian digunakan oleh operator untuk mengumpulkan kartu yang baris pertama dilubangi, diikuti oleh baris ke-2, dan seterusnya.

Radix Sort

Sorting Algorithm

Radix sort adalah algoritma stabil berbasis non-comparison yang menggunakan pengurutan penghitungan dengan sedikit modifikasi untuk mengurutkan bilangan dalam suatu array.

01 Waktu pengurutan algoritma yang linear.

02

Pengurutan algoritma yang stabil. 03

Algoritma pengurutan berbasis Non-comparison. 04

Dapat bekerja untuk kuantitas yang besar 05

Pengurutan element bilangan yang berdasarkan digit

Cara Kerja

Dalam Radix Sort, bilangan diurutkan berdasarkan digit. Mulai dari LSD [Least Significant Digit] to MSD [Most Significant Digit]. Bilangan diurutkan berdasarkan digitnya dengan menggunakan counting sort untuk membantu menjaga stabilitas Algoritmanya.

Langkah Kerja

1. Dilakukanya pencarian untuk bilangan terbesar dalam array atau dalam kata lain bilangan yang memiliki max digit.
2. Digit dari Bilangan terbesar menjadi acuan penyortiran. Untuk setiap pengurutan, Algoritma yang digunakan adalah counting sort algorithm.
3. Counting sort tersebut dimodifikasi, untuk mengurutkan bilanganya berdasarkan digit.

Modifikasi dari counting sort untuk menjalankan algoritma yang sesuai.

1. Ukuran dari suatu array dideklarasikan dengan b, diamana b menunjukan basis dari bilangan dalam suatu array.
2. Daripada menggunakan elemen array sebagai index dalam count array, Digit dari bilanganlah yang digunakan sebagai index dari count array tersebut.
3. Langkah lainya tetap sama seperti tujuan utama dalam counting sort algorithm.

Big O Radix Sort

Jika semisal terdapat d digits di bagian input integers. Radix Sort akan membutuhkan waktu O(d*(n+b)) dimana B adalah basis untuk mewakili urutan angka.

Sebagai contoh, untuk system desimal, b adalah 10. Maka berapakah jumlah d? jika k adalah nilai maksimumnya, maka d akan menjadi $O(\log_b(k))$ Jadi keseluruhan kompleksitas waktu adalah $O((n+b) * \log_b(k))$. Yang dimana dibandingkan dengan comparison-based sorting algorithms, kompleksitas waktu yang didapat lebih besar untuk k berkuantitas besar.

Mari kita batasi k terlebih dahulu. Misalkan $k <= n^c$ dimana c merupakan sebuah konstanta. In that case, Dalam kasus tersebut, kompleksitasnya menjadi O(nLogb(n)). Tapi itu masih tidak dapat mengalahkan comparison-based sorting algorithms.
Bagaimana jika kita membuat nilai b lebih besar?. erapakah nilai b agar kompleksitas waktu menjadi linier? Jika kita menetapkan b sebagai n, kita mendapatkan kompleksitas waktu sebagai O(n). Dengan kata lain, kita dapat mengurutkan array bilangan bulat dengan rentang dari 1 hingga nc jika bilangan tersebut direpresentasikan dalam basis n (atau setiap digit membutuhkan log2(n) bit).