**Simple Search & Binary Search**

Bab 1 : introdusction to algorithms

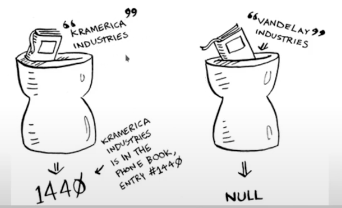
Algoritma adalah sekumpulan instruksi untuk menyelesaikan suatu task atau tugas tertentu. Kita akan membahas sejumlah algoritma yang dipilih karena algoritma nya cepat dan mampu menyelesaikan problem yang cukup menarik.

binary search

Kita akan dihadapi pada suatu kasus dimana harus mencari kata yang diawalai dari huruf k dalam kamus. Pada kasus ini dapat kita pahami bahwa dalam kamus menyimpan sekumpulan kata dalam urutan alfabet, oleh sebab itu kata yang berawalan k tentu saja berada di tengah tidak mungkin diawal ataupun diakhir, jadi kita dapay mulai melakukan pencarian melalui bagian tengah kamus.

Kasus pencarian ini biasa kita kenal dengan istilah search problem atau masalah pencarian, salah satu solusi yang bisa kita gunakan ialah menggunakan binary search.

Binary search adalah algoritma dimana inputan dari algoritmanya dalah suatu list yang terdiri dari sejumlah elemen yang terurut. Ketika elemen yang dicari berada dalam list yang terurut makan binary search akan mengembalikan posisi dari kata tersebut, tetapi jika kata yang dicari tdk ditemukan maka binary search akan mengembalikan nilai null atau kosong.



A simler search (problem) example

saya sedang memikirkan angka 1 sampai dengan 100, kemudian kalian harus menebak angka tersebut dengan jumlah tebakan sesedikit mungkin. Untuk tebakan yang diberikan saya akan beritahu apakah tebakannya terlalu kecil, terlalu besar atau benar.

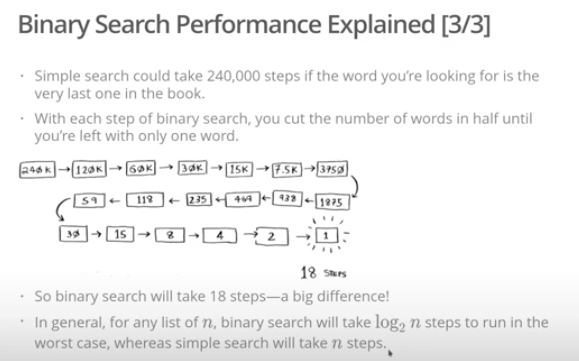
* Simple search way



Algoritma ini dikenal dengan simple search yang mana merupakan metode pencarian sederhana di mana setiap elemen dalam daftar atau array diperiksa satu per satu secara berurutan hingga elemen yang dicari ditemukan atau sampai akhir dari kumpulan data. Ini adalah pendekatan langsung yang efektif untuk set data yang kecil, tetapi bisa menjadi tidak efisien untuk set data yang besar karena membutuhkan jumlah langkah yang proporsional dengan jumlah elemen dalam set data.

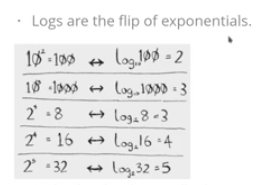
* Binary Search Performance Expalined

Binary search adalah algoritma pencarian yang efisien untuk menemukan nilai tertentu dalam sebuah array yang telah diurutkan. Algoritma ini bekerja dengan membagi set data menjadi dua bagian dan memeriksa apakah nilai yang dicari berada di bagian kiri atau kanan. Proses ini terus berlanjut pada bagian yang sesuai dengan pencarian, mengurangi set data yang harus diperiksa menjadi setengah pada setiap iterasi. Hal ini memungkinkan pencarian dilakukan dengan cepat, terutama pada set data yang besar. Binary search biasanya menggunakan logaritma.



Logarithms

Secara garis besar, logaritma merupakan sebuah operasi invers (kebalikan) dari eksponen atau perpangkatan.



Ketika kamu mencari suatu elemen menggunakan simple search, kemungkinan terburuknya kamu harus melihat dan menghitung setiap elemen satu-satu, dengan kata lain jika kita memiliki 8 buah elemen kita harus melakukan pengecekan sebanyak 8 kali.

Sebaliknya jika kita menggunakan binary search dalam kondisi terburuk kita hanya perlu melakukan pencarian sebanyak log n. sebagai contoh ketika kita mencari list dari 8 elemen kita hanya perlu mencari log = 3, karena 2^3 = 8. Dengan kata lain jika jumlah list sebanyak 8 elemen kita hnya perlu melakukan pengecekan sebanyak 3 kali.

Untuk list sebanyak 1.024 elemen, log 1.024 = 10 karena 2^10 = 1.024 jadi untuk list sebanyak 1.024, kita hanya perlu melakukan pengecekan sebanyak 10 kali.

Binary search hanya bisa digunakan ketika list yang kita miliki sudah terurut. Sebagai contoh, daftar nama pada buku telfon ataupun daftar kata pada kamus, jadi kamu dapat menggunakan binary search untuk menemukannya.

Binary Search In python

Kita akan mulai melihat bagaimana implementasi binary search dengan menggunakan bahasa pemograman python dimana kita akan menampung sekumpulan nilai didalam suatu list.

Low = 0

High = len(list) – 1

Disini menjelaskan tentang konsep indexing dimana posisi pertama memiliki index 0, posisi kedua memiliki index 1 dan seterusnya. Disini akan terdapat suatu fungsi yaitu binary search dimana fungsi ini akan menerima dua parameter, parameter pertama berupa sorted list dan parameter kedua berisi item yang akan dicari dan bilamana item yang terdapat dalam list tadi maka fungsi tersebut akan mengembalikan posisi dari item tersebut didalam list nya,

untuk setiap tahapan pencarian kita akan memantau posisi dari listnya

mid = (low + high) /2

guess = list[mid]

dalam setiap literasinya kita akan memeriksa elemen yang berada pada posisi tengah, oleh karenyanya kita perlu mencari posisi tengahnya seperti pada, selanjutnya elemen yang mau kita periksa adalah list index ke mid yang kita tamping pada variable guess.

Disini nilai mid akan otomatis dibulatkan kebawah oleh python ketika hasil penjumlahan low dan high nya tidak berupa bilangan bulat.

If guess < item:

low = mid + 1

kalau guess nya lebih kecil dari item, maka nilai dari low indexnya bisa kita sesuaikan dengan mid + 1, tetapi jika guess nya lebih besar maka kita perlu mengupdate high indexnya.

Full code:

def binary\_search(list, item):

low = 0

high = len(list)-1

while low <= high:

mid = (low + high) // 2

guess = list[mid]

if guess == item:

return mid

if guess > item:

high = mid – 1

else:

low = mid + 1

return None

my\_list = [1,3,5,7,9]

print(binary\_search(my\_list, 3))

print(binary\_search(my\_list, -1))