Algoritma Searching dalam Keseharian  
Akhtar Abdurrasyid Nitisastra

Problem

Dimas mendapatkan tugas dari atasan untuk mencari CV dari satu nama pegawai yang akan mengikuti *interview* sore ini di berkas arsip. Semua itu telah disimpan di dalam berkas yang dimana nama-nama calon tersebut dikategorikan sesuai ID yang terdiri dari empat digit, dimana dua digit pertama diambil dari nomor huruf pertama nama depan (A —> 01, B —> 02 dst.) dan dua digit kedua diambil dari nomor huruf pertama nama belakang. Dimas diminta mencari nama calon yaitu Panggih Tuflihun. Jika setiap ID dalam berkas itu unik dan satu ID hanya dimiliki oleh satu orang, buatlah minimal 2 algoritma beserta *pseudocode*-nya!

Algoritma

**Nama calon**: Panggih Tuflihun  
 *P —> huruf ke-16, T —> huruf ke-20***ID calon**: 1620

* 1. **Linear Search**: O(n)  
     —> Mencari ID yang diminta dari urutan awal dengan mengecek satu per satu  
       
     ***Pseudocode***:  
     1 Open document and check on the list  
     2 Check the current ID  
     2 ***if*** current ID checked is 1620   
     3 —> Extract file   
     4 ***else if*** current ID is not 1620  
     5 --> look for the next ID (increment by 1)  
     6 Go back to line 2  
     11 **else** quit  
       
     ***Jumlah pencarian maksimal***: 410 pencarian  
     Anggap satu pencarian memerlukan waktu 2 detik;  
     —> Maka waktu yang digunakan dalam algoritma ini adalah **820 detik.**
  2. **Binary Search**: O (log n)—> Mencari ID yang diminta dengan membuka tengah dari list berkas, kemudian menentukan apakah mencari ke kiri atau ke kanan dengan melihat posisi ID yang diminta.  
       
     ***Pseudocode***:  
     1 Open document and check the middle of the list  
     2 Check on the current ID  
     3 ***if*** current ID is 1620  
     4 --> Extract the CV from the ID file  
     5 ***else*** ***if*** current ID is not 1620  
     6 —> look for the next ID to the left side of current position  
     7 Go back to line 2  
     8 ***else if*** current ID is not 1620  
     9 --> look for the next ID to the right side of current position  
     10 Go back to line 2  
     11 **else** quit  
       
     ***Jumlah pencarian maksimal***: 85 pencarian  
     Anggap satu pencarian memerlukan waktu 2 detik;  
     —> Maka waktu yang digunakan dalam algoritma ini adalah **170 detik.**  
     Chart, line chart

     Description automatically generated