Drugi projekat

Potrebno je napraviti mašinu za pretraživanje jednog dokumenta u pdf formatu (search engine). Program prilikom startovanja treba da obiđe pdf dokument, da parsira stranice u njemu i da izgradi strukture podataka potrebne za efikasno pretraživanje. Nakon toga, program omogućuje korisniku da unosi tekstualne upite koji se sastoje od jedne ili više reči razdvojenih razmakom, pretražuje stranice koristeći prethodno kreiranu strukturu podataka i korisniku ispisuje rangirane rezultate pretrage.
Kao test fajl koristiti knjigu *'Data Structures and Algorithms in Python'* (nalazi se u Files/Literatura).

Za maksimalnih 10 poena:

- (4 poena) Rezultati treba da sadrže redni broj rezultata, redni broj stranice, kao i kratak kontekst (isečak iz dela stranice) u kome je tražena reč pronađena. Poželjno je da tražena reč (ili više njih) bude označena u isečku (npr. bojenjem konzolnog ispisa).
- (2 poena) Implementirati rangiranje rezultata pretrage tako da na rang rezultata utiče broj pojavljivanja traženih reči na stranici uz korišćenje proizvoljnih struktura podataka.
- (2 poena) Ukoliko korisnik unosi upit sastavljen od više reči, vršiti rangiranje stranica tako da pojavljivanje svake od reči utiče na sveukupno rangiranje određene stranice. U ovom slučaju, ne treba insistirati na prisustvu svake od reči u rezultatima, ali bi trebalo bolje rangirati rezultate u kojima se pojavljuju sve reči.
- (2 poena) U cilju podrške traženim operacijama, obezbediti konzolni meni. Korisniku se nude opcija da započne pretragu. Posebne vrste pretrage ne treba da budu dodatne opcije u meniju već korisnik samim načinom unosa sugeriše koja vrsta pretrage treba da se izvrši (uvođenjem navodnika ili zvezdice, navođenjem više reči...).
- Umesto pdf dokumenta, za formiranje struktura se mogu koristiti već parsirani txt fajlovi (iz Files/Projekat 2).

Za maksimalnih 17 poena

Za maksimalnih 17 poena potrebno je realizovati sve tačke za 10 poena i još:

- (2 poena) Na rangiranje, osim broja pojavljivanja traženih reči na stranici, treba da utiče i broj veza sa drugih stranica kao i broj traženih reči na stranicama koji sadrže vezu na traženu stranicu. Veze ka stranicama se mogu prepoznati na osnovu napomena u tekstu (npr. 'See page 136', '... on page 87', see pages 34 and 35').
- (3 poena) Za organizovanje stranica koristiti graf.
- (2 poena) Za efikasnu pretragu reči na stranici koristiti strukturu podataka trie.

Za više od 17 poena potrebno je obezbediti učitavanje sadržaja iz pdf fajla uz pomoć neke od biblioteka za rad sa pdf dokumentima.

Za maksimalnih 21 poen

Za maksimalnih 21 poena potrebno je realizovati sve tačke za 10 poena, sve tačke za 17 poena i još:

- (1 poen) Serijalizacija: U cilju efikasnijeg izvršavanja operacija, omogućiti serijalizaciju formiranih struktura podataka kako se ne bi trošilo vreme za njihovo rekreiranje prilikom svakog pokretanja.
- (2 poena) Potrebno je podržati ispravnu upotrebu logičkih operatora AND, OR i NOT prilikom formiranja upita uz kombinovanje operatora.
  Primeri:
  python AND sequence
  dictionary NOT list
  python OR dictionary NOT word

- (1 poen) Obezbediti paginaciju rezultata. Ograničiti broj ispisa po stranici na N (rezultati od 1 do N) uz nuđenje opcije za ispis sledećih N rezultata (od N+1 do 2N) itd.

Za više od 21 poen

- (1 poena) Upotreba fraza. Fraza se navodi pod navodnicima. U rezultatima se prikazuju (uz rangiranje) stranice u kojima se navedeni delovi fraze pojavljuju uzastopno u istom redosledu.
- (1 poena) Implementacija predlaganja alternativnih ključnih reči za pretragu *(did you mean)*. Ukoliko rezultata pretrage nema (nema rangiranih stranica) ili se zadati upit pojavljuje na malom broju stranica, ponuditi korisniku da zadati upit zameni sličnim, popularnijim upitom.
- (1 poen) Implementirati grupisanje operatora AND, OR i NOT pomoću zagrada.
- (1 poena) Autocomplete. Odabirom ove opcije, korisniku se nudi nekoliko popularnih završetaka zadatog upita, npr. ako korisnik unese *fun** ponuđene opcije mogu biti *functionality* i *function* (u case insensitive režimu).

Poeni za dodatne funkcionalnosti se mogu osvojiti samo u slučaju implementacije zadataka za više od 22 poena.
. Za većinu dodatnih funkcionalnosti potrebno je istražiti mogućnosti drugih biblioteka za rad sa pdf dokumentima.
Primeri:

- 1 poen - Čuvanje rezultata pretrage u obliku pdf-a. Izdvojiti stranice prvih 10 rezultata i objediniti ih u poseban pdf fajl.
- 2 poena - Označiti (pdf highlighting) bojom pronađene ključne reči (kao u drugom pasusu) u pdf dokumentu sačinjenom objedinjavanjem stranica.
- Druge funkcionalnosti koje predmetni profesor ili asistent odobre.

Dodatna objašnjenja:

Logički operator AND zahteva prisustvo svih navedenih reči u stranicama koje su u rešenju. Sve navedene reči bi trebalo u jednakoj meri da utiču na ukupan rang.
Logički operator OR zahteva prisustvo bar jedne od navedenih reči u stranicama koje predstavljaju rešenje.

Logički operator NOT se u ovom slučaju smatra BINARNIM dakle, predviđa se upotreba u obliku 'reč1 NOT reč2' gde se pojavljivanje reči *reč1* zahteva u rešenju dok se *reč2* u rešenju ne sme pojaviti.

Opšte informacije o slanju i upload-u zadatka:

- Zadatak nosi 25 poena.
- Smestiti sve fajlove zadatka u **folder** pod nazivom projekat2_sv_XX_YYYY gde se umesto XX_YYYY navodi broj indeksa - broj upisa i godina upisa (primer: projekat2_sv_02_2023)
- Ubaciti fajl u **zip** arhivu i nazvati je isto kao i zadatak (projekat2_sv_XX_YYYY.zip)
- Uploadovati zip arhivu kao assignment na enastavu.
- Ukoliko bude problema sa uploadom, možete u predviđenom roku poslati zip na email adresu predmetnog asistenta.

Rok za slanje arhive je 28.06.2024.